TWO-DIMENSIONAL SHAPE DESCRIPTION

Thomas W. Rauber

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
&
UNINOVA - Intelligent Robotics Center
Quinta da Torre, Monte de Caparica, Portugal
E-mail: tr@fct.unl.pt

Technical Report: GR UNINOVA-RT-10-94

This technical report is a chapter of the following manuscript:
Inductive Pattern Classification, -Methods-Features-Sensors-,
Thomas Walter Rauber
Ph.D. thesis, Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia,
Lisboa, Portugal, 1994.

Lisboa, 1994

CHAPTER 5 TWO-DIMENSIONAL SHAPE DESCRIPTION

This chapter is dedicated to a novel shape descriptor for binary two-dimensional (2-D) visual patterns, called *UNL Fourier Features*. It is based on a normalizing parametric coordinate transformation of the pattern, denoted as *UNL transform*. It is assumed that the pattern can be composed of a finite set of parametric curves which is possible in most cases.

First an overview of existing shape description techniques is given. Then the new method is introduced, its theory in the analytic case and the methods to apply it to concrete forms, for which an analytic description of its composing curves is not explicitly given.

Experiments corroborate the usefulness and general purpose applicability of the proposed method. Not only the form of objects from the context of 2-D robot vision can be characterized by the UNL Fourier Features, but also any shape that satisfies the constraints of the application domain. The recognition of handwritten characters is a possible field of application which will be proved by practical experiments.

5.1 Shape Description Methods for Binary Patterns

In order to create some background for the introduction of the novel method, some existing techniques for the representation and description of 2-D objects are reviewed. The list of the procedures listed here is by far not a complete summary, but serves as an orientation within this specific topic of artificial vision. Each of the methods is shortly described with an emphasis on the respective advantages and/or drawbacks.

Both [Pavlidis, 1978] in an early survey over existing shape description methods and [Gonzalez, 1987] distinguish two different families of techniques, those describing the *interior region* of the object and those describing the *boundary*. This distinction is already a constraint by itself for the descriptor. For the accessibility to both domains, it assumes that the object must posses both an interior area and a boundary that delimits the object. Many methods assume furthermore that the boundary of the object is closed. This is normally not the case for patterns like handwritten characters if they are composed only of strokes which do not have a measurable thickness. Also nested objects pose an insurmountable obstacle to the majority of the shape descriptors. A hole inside a closed boundary of an object cannot be described by the same descriptor as the boundary itself. The only remedy to overcome this problem is to use a hierarchical organization of descriptors to characterize a nested shape. Generally speaking, there is no method suitable for the description of any kind of pattern.

[He, 1991] gives also a short overview for 2-D shape classification techniques in the introduction of a paper which uses hidden Markov models for the description of partial sequences of the boundary of an object.

A common quality requirement can be defined for all shape descriptors. The *shape* or *form* of the object is an attribute that must not change when the original object is submitted to a certain set of affine geometric transformations (Figure 5.1), or an arbitrary combination of the same. A 2-D shape descriptor should be insensitive to:

- Translations
- Scale changes (uniform in both the x-coordinate and the y-coordinate)
- Rotations

This axiom for shape descriptors implies that the descriptor is able to perform a normalization for the different appearances in which the object may occur. The actual instantiation of the object must be mappable to an isomorphous prototype of the pattern.

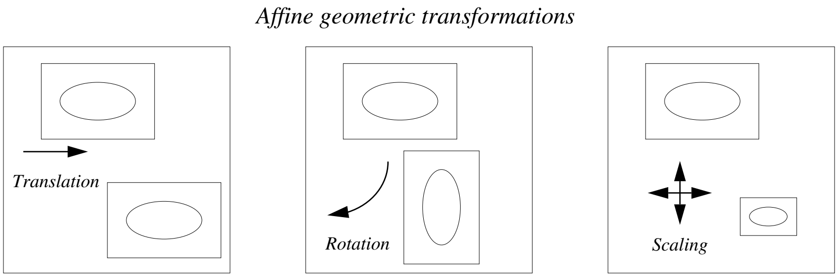


Figure 5.1 The form of an object is insensitive to the three affine geometric transformations: translation, rotation and uniform change of scale.

Description is the basis for classification. Ideally, the descriptors should be quantitative values which can be processed by a digital computer. It is then possible to build a pattern recognition system on top of the describing shape features.

In order to evaluate the improvement that the UNL Fourier Features represent, compared to already existing 2-D shape representation methods, a benchmark pattern is proposed. The stylized head of a toy robot in Figure 5.2 will be used for this purpose. Parts of it could be modelled by a region-based technique, like the head with eyes, nose and mouth. The antenna however are only representable as curves without a defined thickness.

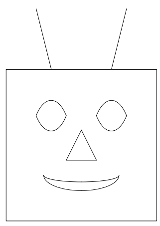


Figure 5.2 Test pattern for 2-D shape descriptors.

5.1.1 Boundary Descriptors

Edge detection in image processing is an important preprocessing step. Its objective is the extraction of boundary points of the objects in the scene. The transitional zones of the image where a change between regions of interest and background takes place, contain most of the information of the scene. A human being is able to recognize a form often by merely regarding its silhouette. A shadowplay is an example of a pattern recognition process where the viewing public extracts all visual information from only the boundaries of the participating objects and persons.

Some of the techniques which are based exclusively on the boundary of the object, are described in the following.

5.1.1.1 Chain Codes

One compression schema for the boundary of a shape are chain codes [Freeman, 1961], [Gonzalez, 1987]. Starting from one boundary point, the next point is searched in one of a small set of possible directions, based on the neighborhood model of the image, 4-connectivity or 8-connectivity [Gonzalez, 1987]. Two, respectively three bits are needed for recording the sequence of directional changes when following the curve. Chain codes are usually employed for the encoding of the boundary of binary objects in image processing systems. As the advantages (+) and drawbacks (-) of this boundary descriptors one can mention:

- \+ Direct obtainment of numerical encoding of the boundary. A quantitative description of the boundary of the object can be derived from the binary image by the chain code technique.
- \+ A compact representation of the boundary. Instead of storing the absolute coordinates of the boundary points, a sequence of directional changes is used. This saves a lot of storage memory.
- \+ Normalization possible. The chain code is invariant to translations of the object. A normalization relative to rotations and scale changes is feasible.
- \- The method is normally conceived only for closed boundaries.
- \- Transformations of the pattern (rotation, scale) change the pattern itself. This signifies that the normalized chain code obtained from the transformed pattern is not equal to the chain code of the original pattern.
- \- Noise sensitivity. For the purpose of direct matching of the chain code in a classification task, noise must be reduced to a minimum. If for instance a convexity of the object is lost because of an illumination disturbance, the chain code will be completely different in length and morphology. Therefor a matching will surely be impossible.
- \- No nested processing. Objects inside other objects must be described by a separate chain code. The global structure of a nested form cannot be characterized by a single chain code. This is also valid for the test pattern of Figure 5.2.

5.1.1.2 Polygonal Approximations

The idea behind this technique is to describe an eventually complex contour by a sequence of straight lines. By varying the number of the approximating lines, the coarseness of the resulting polygonal model of the original pattern can be controlled. In [Fu, 1987] and [Gonzalez, 1987], a series of different approximation techniques is presented. [Nasrabadi, 1991] uses the split and merge method proposed by [Pavlidis, 1977a], [Pavlidis, 1977b], [Pavlidis, 1982] as the feature model for a classifier based on a Hopfield neural network. Polygonal approximations

- \+ achieve a high compression rate for regular shapes.
- \+ can be used for the incomplete matching of shapes. This permits the recognition of objects which are partially occluded.
- \- allow no nested processing of curves inside other curves.
- \- are sensitive to disturbances of the contour.
- \- are not invariant to rotations and scale changes.

Also here the complexity of the pattern in Figure 5.2 does not allow to represent the totality of the particular sub-patterns in a single global descriptor.

5.1.1.3 Signatures

The commonly used signature is the distance from the centroid of a closed boundary to the boundary points [Fu, 1987], [Gonzalez, 1987]. Generally speaking, signatures are one-dimensional representations of two-dimensional curve shapes. One-dimensional moments and Fourier descriptors [Fu, 1987], [Gonzalez, 1987] often use signatures as the intermediate representation, before calculating the final features. For the special case where the boundary distance function is the signature, the following can be stated:

+ Simple shapes are concisely represented by signatures.
+ Normalization for scale changes is relatively easy, translation insensitiveness, rotation results in a cyclic shift of the function values.
- No nested boundaries allowed.
- Concave objects generate ambiguous distance measures if the distance is sampled along a constant increment of the *angle*. If the distance is sampled along a constant *curve length*, concave objects can be represented, but the distance function is not 2π-periodic. In this case the signature is rotation sensitive.

For the test pattern of Figure 5.2 the following can be observed: the eyes, nose, mouth and the boundaries of the head can be represented as signatures. The antenna and most important the figure as a whole are not mappable to a single signature.

5.1.1.4 Fourier Descriptors

The first elaborated definition of the Fourier descriptor (FD) was given by [Zahn, 1972] and later refined by [Persoon, 1977]. The technique of FDs can claim to be a real shape feature of a plane closed curve. For many boundary patterns the method of FDs can satisfactorily characterize the form of the pattern. Encoding, reconstruction and classification of 2-D curves is possible with Fourier descriptors. The original definition of [Zahn, 1972] is given here.

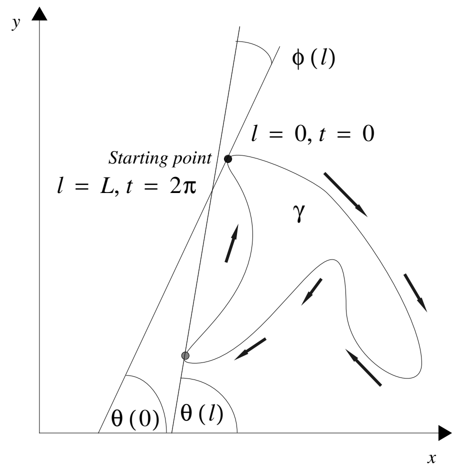


Figure 5.3 Angular direction $\theta(l)$ and cumulative angular function $\phi(l)$ as the basis for Fourier descriptors.

The *arc length* l along the curve γ can be calculated from a parametric representation of the curve coordinates $z(t) = (x(t), y(t))$, where the parameter t varies between 0 and a maximum value T and the curve is assumed to be smooth, i.e. the derivative $\dot{z}(t)$ exist for all $t \in [0, T]$.

$$l(t) = \int_0^t \| \dot{z}(\tau) \| d\tau, \text{ where } \| . \| \text{ is the Euclidean norm.}$$

The parametric representation of the curve can equivalently be expressed by the parameter l: $z(l) = (x(l), y(l))$. The angular direction of γ at arc length l is $\theta(l)$, see Figure 5.3. The *cumulative angular function* $\phi(l)$ sums the absolute angle $\theta(l)$ along the whole curve. For a smooth closed curve the relation $\phi(l) = \theta(l) - \theta(0)$ holds.

The arc length l is the period for the periodic function $\phi(l)$. Since the ideal period for a Fourier transforms is 2π, $\phi(l)$ is normalized, using the new parameter t, that varies between 0 and 2π. In order to compensate for the cumulative angle -2π that is accumulated with $\phi(l)$ when the curve is scanned once, an additive term is joined to the normalized form to obtain the periodicity condition $\phi^*(0) = \phi^*(2\pi) = 0$.

Normalized cumulative angular function: $$\phi^*(t) = \phi\left(\frac{Lt}{2\pi}\right) + t \qquad \text{EQ 5.1}$$

Finally the Fourier expansion of the function $\phi^*(t)$ yields the Fourier descriptors (A_k, α_k) :

Fourier expansion: $$\phi^*(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k) \qquad \text{EQ 5.2}$$

Fourier descriptors: (A_k, α_k), $k = 1 \ldots \infty$

The A_k are the *harmonic amplitudes* of the Fourier expansion and the α_k are the *phase angles*. It can be verified that the FDs are invariant to the translation of the shape. The A_k are not affected by rotations or a different starting point. The normalization against changes in the scale of the shape is achieved by dividing each A_k by A_1 which is normally the largest of the harmonic amplitudes.

In practice it turns out that only the harmonic amplitudes A_k are appropriate for the classification of the shape [Gonzalez, 1987]. The noise introduced by the discrete nature of the image grid reduces the discriminative properties of the phase angles α_k to a minimum.

Particular problems with the behavior of the FDs are not elaborated here. Only the ideal case where the shape is a closed, smooth curve is considered. FDs can as well be obtained from polygons and line patterns that do not possess an interior region [Persoon, 1977].

+ FDs capture the shape of the object in a feature vector ideal for machine supported classification.
+ FDs are easy to normalize.
+ FDs are information preserving. The original curve pattern can be reconstructed from the Fourier expansion. This is true due to the very nature of the Fourier transform.
- Curves must be closed. Problems arise when the curve has any discontinuity. Heuristic preprocessing must be applied to the curve in order to close it.
- No nested processing of curves is possible.

Once again it is not possible to find a single set of Fourier descriptors that can store the pattern of Figure 5.2. It is not possible to describe the whole pattern by a single, global set of FDs. It can be expected that the head, the eyes, nose and mouth by themselves are easily characterized by their respective FDs. For the antenna, the line pattern method [Persoon, 1977] can be applied. The curve is traced from the bottom to the top and then once again from the top to the bottom, thus implicitly assuming that the line has a non-zero thickness.

5.1.2 Region Descriptors

We pass now to a series of methods that represent the shape not by its boundary but by its interior region. The first commonsense conclusion is that objects without an interior region are nor representable by this group of shape descriptors. It fact it can be observed in the reviewed literature that all objects supposed to be characterized by region descriptors had a non-zero thickness. This is true for the method of *moments* and the *Log-Polar transform* which will be presented here.

5.1.2.1 Moments

The concept of moments as invariant image features was introduced by [Hu, 1962]. [Reiss, 1991] revised the theory of the *Fundamental Theorem of Moment Invariants*. A representative application of moments together with FDs to recognize airplane silhouettes can be found in [Reeves, 1988]. See also [Teh, 1988] and [Khotanzad, 1990] for an excellent treatment of the subject. Principally we will refer to the fundamental presentation of moments given in [Gonzalez, 1987].

The analytic definition for a moment of order $p+q$ and a continuous 2-D function $f(x, y)$ is given as

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x, y)\, dx dy \qquad p, q = 0, 1, 2, \ldots \qquad \text{EQ 5.3}$$

For the case of the discrete image matrix $f(x, y)$, the moments of order $p+q$ become

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \qquad \text{EQ 5.4}$$

The *central moments* of a digital image are defined as

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y), \qquad \bar{x} = \frac{m_{10}}{m_{00}}, \qquad \bar{y} = \frac{m_{01}}{m_{00}} \qquad \text{EQ 5.5}$$

The *normalized central moments* are derived from the central moments as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \qquad \gamma = \frac{p+q}{2} + 1, \qquad p + q = 2, 3, \ldots \qquad \text{EQ 5.6}$$

Finally for order 2 and 3, seven *invariant moments* ϕ_i, $i = 1 \ldots 7$ can be derived that are insensitive to the affine transformations translation, rotation and uniform changes in scale. For the explicit definition of the seven ϕ_i refer to [Hu, 1962] or [Gonzalez, 1987].

Like the 2-D Fourier transform and the Complex-Log transform which will be characterized below, moments are image oriented features. Basically one can state that the input for moment invariants are pixel values $f(x, y)$, sampled at a certain image point (x, y). Moment invariants are not generated from the analytic information of the object geometry, but rather by an *instantiation* of an object on a 2-D pixel matrix. Ultimately the features are calculated from the pixel values of the image.

For the test pattern of Figure 5.2, the method of moment invariants is only partially suited because the pattern consists only of fine lines. In all classification experiments of the reviewed literature that were based on moment invariants, the patterns had an interior region. This suggests the empirical conclusion that moments are not appropriate to describe patterns where only line oriented shape information is available.

Practical classification experiments will corroborate that moment invariants are no suitable shape descriptors for line patterns.

5.1.2.2 Log-Polar transform (Complex-Log transform)

For the following region-oriented 2-D shape descriptors and later for the definition of the UNL transform it is advantageous to define an image coordinate as a complex number. This sometimes allows a more concise formalism. Hence a Cartesian point (x, y) is represented as a complex number $z = x + jy$.

The mathematical basis for the Log-Polar transform (sometimes also denoted as Complex-Log transform) is the property of the logarithm of a complex number:

$$ln(z) = ln(x + jy) = ln(|z|) + j\theta + 2jk\pi \text{ where } k \text{ is an integer number.} \qquad \text{EQ 5.7}$$

The Euler relation $z = |z|e^{j\theta}$ is used here, with $|z| = \sqrt{x^2 + y^2}$ and $\tan\theta = \frac{y}{x}$.

Normally only the principal part of the logarithm is used ($k = 0$). The result of the logarithm of a complex number is a new complex number, thus defining the Log-Polar transform (LPT):

$$\text{Log-Polar transform} \qquad LPT\colon (x, y) \to (ln(|z|), \theta) \qquad \text{EQ 5.8}$$

The mapping of the points is done from the Cartesian coordinate system to the polar coordinate system. From biology, an analogy to the topology inside the human visual system is established

[Schwartz, 1977], [Schwartz, 1980a], [Schwartz, 1980b]. The points are mapped from the *retinal plane* to the *cortical plane*.

The consequences from the property of the logarithm of a complex number are that a rotation of z by an angle $\Delta\theta$ to $z' = |z|e^{j(\theta + \Delta\theta)}$ results in the mapped number $ln(z') = (ln(|z|), \theta + \Delta\theta)$. Hence a *rotation* of the original point in the Cartesian coordinate system is transformed to a *translation* of the ordinate in the polar coordinate system. A multiplication of z by a scalar a results in a translation of the abscissa of the polar coordinate. Thus also a *scaling* results in a *translation*: $z' = a|z|e^{j\theta}$, $ln(z') = (ln(a|z|), \theta) = (ln(|z|) + ln(a), \theta)$.

The *translation* of the original Cartesian point $z' = |z| + \Delta z$ is compensated for by placing the origin of the Cartesian coordinate system into the centroid of the pattern, before the Log-Polar transform is applied.

In [Massone, 1985] the Log-Polar transform is used for the recognition of 2-D objects. A region of interest of the image is transformed from the Cartesian to the polar coordinate system. A non-uniform sampling structure is placed over the center of gravity of the object. The sampling process is determined by several heuristics which limit the general purpose applicability of the method. The final recognition is done by template matching. A template of the Log-Polar transform of all object classes is stored. In the recognition phase the Log-Polar transform of the unknown object is correlated with all classes and the class with the highest correlation score is attributed to the unknown shape.

[Wechsler, 1988] used also the Log-Polar transform as a basis for the classification of 2-D patterns. Here, especially the problems associated with the Log-Polar transform are mentioned.

- Since the sampling schema of the objects has a decreasing resolution from the center to the edges, information is lost. An inverse transform from the polar coordinate system back to the Cartesian coordinates exhibits a *smearing* effect. Details of the image far away from the center of the sampling structure become unclear.
- One of the biggest problems of the Log-Polar transform is the high sensibility to a slight misalignment of the center of gravity. If for the same object the center of gravity is computed differently, the resulting images in the polar coordinate system can be totally distinct. This results normally in poor recognition accuracy.
- The Log-Polar transform is not really scale invariant. A *stretching* effect can be observed when the object in different sizes is transformed to the polar coordinate system. If a shape is scaled by a factor a, the offset of the abscissa of a particular point z of the shape will always be $ln(a)$, independently of the point z. The abscissa of the transformed coordinate $ln(z)$ however depends always on the point z itself.

Also the Log-Polar transform works on patterns that posses an interior region. The test pattern of Figure 5.2 could only be submitted to this 2-D shape descriptor if it had an interior region and the antenna were missing or had a finite thickness. [Reber, 1987] applies the Log-Polar transform to the recognition of characters. Also here it can be observed that the objects have a non-zero thickness.

The UNL transform and the Log-Polar transform have some characteristics in common. Later, an explicit comparison of the two techniques will be presented in order to emphasize the distinctiveness of the UNL transform.

5.1.2.3 Two-dimensional Fourier transform

The Fourier transform is one of the most important tools for signal analysis. Not only visual patterns, but any physical signal that is of periodic nature, normally provides more information if it is analyzed by the Fourier transform. A signal from the *space* or *time* domain is transformed to the *frequency* domain. For instance in voice recognition systems the acoustic signal is almost always analyzed regarding its frequency composition.

Here, we will only shortly mention the definition of the 2-D Fourier transform and highlight its properties that are essential for the definition of the UNL Fourier Features. For a good introduction to the Fourier transform in the context of image analysis refer for instance to [Gonzalez, 1987].

The analytical *Fourier transform* $\mathcal{F}$ of the continuous and integrable 2-D function $f(x, y)$ is

$$\mathcal{F}\{f(x, y)\} = F(u, v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\, e^{-j2\pi(ux + vy)}\, dxdy \qquad \text{EQ 5.9}$$

In general, the Fourier transform $F(u, v)$ of the real function $f(x, y)$ is a complex function $F(u, v) = |F(u, v)|e^{j\phi(u, v)} = \Re(u, v) + j\Im(u, v)$. $|F(u, v)|$ is called the *Fourier spectrum* and $\phi(u, v)$ is called the *phase angle* of the Fourier transform.

If the continuous signal $f(x, y)$ is quantized into $N \times N$ discrete values, the *discrete 2-D Fourier transform* is defined as

$$\mathcal{F}\{f(x, y)\} = F(u, v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x, y)\, e^{\frac{-j2\pi(ux + vy)}{N}} \qquad \text{EQ 5.10}$$

In practice, the discrete Fourier transform is implemented as a Fast Fourier Transform (FFT) which reduces the complexity of the calculus from $O(N^2)$ to $O(N\log_2 N)$. The condition $N = 2^n$ must be satisfied, where n is a positive integer number.

The most important properties of the 2-D Fourier transform that interest us here are the *rotation* and *translation* property:

Lemma 5.1 **Rotation property of the 2-D Fourier transform**
If the signal $f(x, y)$ is rotated by an angle $\Delta\theta$, the Fourier transform $F(u, v)$ is rotated by the same angle $\Delta\theta$.

Lemma 5.2 **Translation property of the 2-D Fourier transform**
If the signal $f(x, y)$ is shifted by an offset $\Delta z = (\Delta x, \Delta y)$, the Fourier transform $F(u, v)$ is multiplied by the term $e^{-j2\pi(u\Delta x + v\Delta y)}$. This signifies that the Fourier spectrum $|F(u, v)|$ is invariant to shifts of the signal $f(x, y)$:
$\left|F(u, v)\, e^{-j2\pi(u\Delta x + v\Delta y)}\right| = |F(u, v)|$

In image analysis, the Fourier transform plays an important role for the recognition of objects inside a scene. It must be mentioned that ideally the signal $f(x, y)$ should be periodic with a period of 2π. In practice we deal with a finite rectangular region where the signal is zero outside this region. It is then simply assumed that the image extends infinitely outside the border of sampled region.

For the use of the Fourier transform as a shape descriptor, the two above properties lead to the following facts:

i.) The Fourier spectrum is invariant if an object is translated inside the image.

ii.) The Fourier transform (spectrum and phase angle) cannot be used directly as a shape descriptor when an object is rotated inside a scene, since the Fourier spectrum rotates by the same angle as the object.

iii.) If the whole image is analyzed by the Fourier transform, also a change in the scale of an object inside the image changes both Fourier spectrum and phase angle.

The most important statement for the definition of the UNL Fourier Features that will be defined later is:

Lemma 5.3 **Invariance of the Fourier spectrum**
The Fourier spectrum $|F(u, v)|$ of an image is invariant if for the coordinates x or y

A.) Only cyclic shifts occur

B.) There are no changes at all

In Figure 5.4 a situation is depicted where three patterns are generated from each other by a cyclic shift of the x-coordinate. The y-coordinate of the pattern points remains unchanged. Consequently the Fourier spectrum of the three patterns will be identical, conforming to Lemma 5.3.

It is also noteworthy to mention that normally an image $f(x, y)$ is submitted to the Fourier transform where the values of the $f(x, y)$ consist of discrete grey values or other quantized components of a continuous intensity value. It is however perfectly viable to transform an image by the Fourier transform where the values are binary. Even pictures which are composed by only thin lines, like in Figure 5.4 can be analyzed by a 2-D Fourier transform. In this case the values of the image are $f(x, y) = 1$ where a point of a line exists at (x, y) and $f(x, y) = 0$ otherwise. An effect of the analysis of line pattern images by the Fourier transform is that very sharp transitions between pattern points $f(x, y) = 1$ and background points $f(x, y) = 0$ provoke large amplitudes in the higher frequency spectrum of the Fourier spectrum. It can be expected that the discrete nature of a digital image will provoke noise that sometimes outweighs the useful information, especially in the high frequencies. Experiments will corroborate the presumption that only the lower frequencies of the Fourier spectrum are good features for classification purposes.

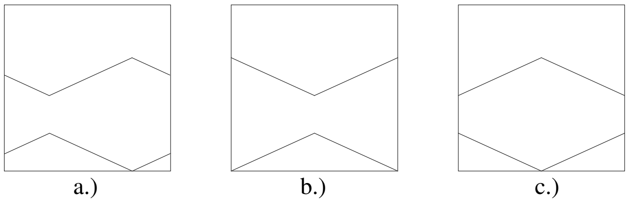


Figure 5.4 Invariance of the Fourier spectrum. From the image in a.) to the images in b.) and c.): In the x-coordinate a cyclic shift occurs. The y-coordinate is not changed.

Speaking in terms of a feature based supervised pattern recognition problem, we still have to define what are the features that serve as the basis for the classification process. In the discrete case the Fourier transform of the 2-D image $f(x, y)$ generates another 2-D image $F(u, v)$. Each of the points of this image $F(u, v)$ has a continuous numerical value. Problems associated with that approach are:

i.) The number of features is extremely high. The number of features grows quadratically with the resolution N of the sampling grid.

ii.) The higher the resolution, the more correlated are neighboring points in $F(u, v)$. With a high resolution it is for instance very probable that the neighboring values $F(2, 5)$ and $F(2, 6)$ do contain the same information for the purpose of classification.

iii.) The feature selection of such a large number of features poses a nontrivial problem. Some techniques for feature selection of continuous values presented in [Devijver, 1982] will be used for the reduction of the size of the feature set to a reasonable dimension.

5.2 UNL[1] Transform and UNL Fourier Features

In this section an alternative method for the description of two-dimensional curve patterns will be presented ([Rauber, 1992b]). First, the necessary nomenclature from the area of parametric curves is given. Then, the analytic theory of the method is introduced. In the discrete case of a digital image, approximative calculi must be used to obtain the shape descriptors since the explicit analytic definition of the patterns is mostly missing. A comparison of the UNL transform to its "nearest neighbor", the Log-polar transform is given, since both techniques share some concepts. Classification experiments will be carried out that prove the general purpose applicability of the method.

5.2.1 Introduction

The overall goal is to find 2-D shape features that are invariant to the affine transformations of a pattern depicted in Figure 5.1 on page 2: translation, rotation and scale changes. The shape descriptor must compensate for these transformations, it must map all possible instantiations of a shape to the same prototype representation. Ideally, this representation should contain as much global information as possible about the shape. Even similar shapes should be distinguishable by a classifier based on the shape descriptor.

The basic idea of the UNL Fourier Features is to achieve exactly this goal. For any instance of an object that was translated, rotated or modified in size, a representation is sought that is always identical, see Figure 5.5.

The constraint for the family of patterns that can be described by the proposed technique, is the following condition: The pattern must be a compound of a finite number of smooth parametric curves. This condition is true for most patterns. A pattern like in Figure 5.2 on page 2 belongs to this group, as well as a handwritten digit, in the form of a closed or open curve. Region-like patterns can equivalently be represented by their boundaries without loss of information.

Texture-like patterns are not suited for the representation by a set of parametric curves, since the details of the texture are very complex and are better described by a local primitive, the regular shapes of the global texture.

The affine geometric transformations applied to the prototype of a curve pattern, namely translation, rotation and scale changes are mathematically well defined operations. The idea behind the UNL Fourier Features is to apply the inverse mathematical operations in order to obtain the original prototype.

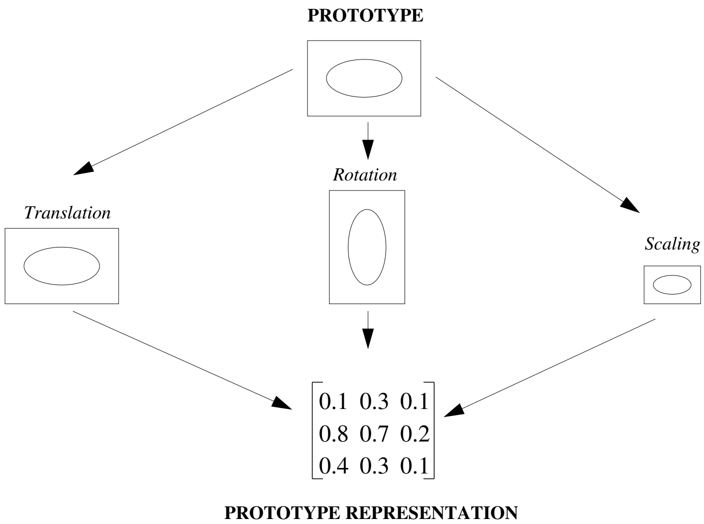


Figure 5.5 General idea of a shape descriptor. An object can occur in different instantiations. From these instantiations an invariant quantitative representation of the prototype must be found.

1. **U**niversidade **N**ova de **L**isboa

The first step of this normalization process compensates for translations and scale changes. It is called the UNL transform. Rotations of the original pattern will result in cyclic shifts of the transformed pattern. The input of the UNL transform is a set of parametric curves. The output is another set of parametric curves. The transform maps from Cartesian to polar coordinates, just like the Log-polar transform. The essential difference however is that the UNL transform maps entire curves and the Log-polar transform maps single points. Besides, the mathematical operation is different.

Like we have seen in section 5.1.2.3, the two-dimensional Fourier transform has a Fourier spectrum that is insensitive to cyclic shifts of the pattern. The Fourier spectrum of the UNL transformed curve pattern constitutes the UNL Fourier Features. The final shape features are defined.

5.2.1.1 Parametric Curves

The input for the shape descriptor are patterns that are composed of a finite set of smooth parametric curves. In order to familiarize the reader with the subjects, some introductory words are said about parametric curves in the n-dimensional Euclidean space in general and in 2-D in particular.

Definition 5.1 **Parametric Curves**

A *parametric curve* in the real Euclidean space $\boldsymbol{R}^n$ is a map $\gamma: I \to \boldsymbol{R}^n$
where I is an open interval in $\boldsymbol{R}$, $I = (a, b)$ where
$\gamma(t) = (\gamma_1(t), \ldots, \gamma_n(t))$
and each function γ_i has derivatives of all orders, for all $t \in I$. Such a γ_i is called *smooth*.
The variable t is called a *parameter*: the point $\gamma(t)$ has parameter value t.
The vector $\dot{\gamma}(t) = (\dot{\gamma}_1(t), \ldots, \dot{\gamma}_n(t)) = d\gamma/dt$ is called the *velocity vector* of γ at t.
In 2-D, a parametric curve is a map in $\boldsymbol{R}^2$, i. e. $z: I \to \boldsymbol{R}^2$, $z(t) = (x(t), y(t))$

Definition 5.2 **Image (trace) of a parametric curve**

The set of points in $\boldsymbol{R}^n$ $C = \gamma(I)$ is called the *image* or *trace* of the curve. In fact, in order to visualize a curve, it is necessary that the curve points have a finite thickness.
The trace of a curve can be though of as the *materialization* or *instantiation* of the curve.
In the case of a 2-D curve in a discrete image, the trace of a curve are the visible pixels.
Here, the points of the curve have implicitly a finite thickness.
In $\boldsymbol{R}^2$ the image or trace of a curve is $I(z(t)) = \{ p \in \boldsymbol{R}^2 | \quad z(t) = p, \quad \forall t \in (0, 1) \}$

The boundaries a, b of the interval I can easily be normalized to $[0, 1]$ by using the parameter transformation $\varphi(t)$ such that $\varphi(a) = 0$ and $\varphi(b) = 1$. As a parameter transformation, $\varphi(t)$ must 1.) be arbitrarily often differentiable and 2.) its first derivative must be non-zero for all $t \in I$. These conditions are satisfied by $\varphi(t) = (t-a)/(b-a)$ (proof omitted).

Figure 5.6 shows some examples for parametric curves $z(t) = (x(t), y(t))$ in 2-D. The parameter t is normalized to the interval $I = (0, 1)$.

$z_1(t) = (0.1t, -0.4t)$ $\qquad$ $z_2(t) = (0.5t, 0.5(t-0.5)^2)$ $\qquad$ $z_3(t) = (\cos t, \sin t)$

Figure 5.6 Examples for parametric curves in 2-D.

5.2.1.2 Centroid of Parametric Curves

The centroid $O = (O_x, O_y)$ of a single parametric curve $z(t)$ can analytically be calculated by the following formula:

$$O = (O_x, O_y) = \frac{\int_0^1 z(t) \, \| \dot{z}(t) \| \, dt}{\int_0^1 \| \dot{z}(t) \| \, dt} \qquad \text{EQ 5.11}$$

The common centroid $O = (O_x, O_y)$ of n parametric curves $z_i(t)$, $i = 1 \ldots n$ is

$$O = (O_x, O_y) = \frac{\sum_{i=1}^{n} \int_0^1 z_i(t) \| \dot{z}_i(t) \| dt}{\sum_{i=1}^{n} \int_0^1 \| \dot{z}_i(t) \| dt} . \qquad \text{EQ 5.12}$$

5.2.1.3 Discrete Case

In the discrete case of a digital image it makes not much sense to define a pattern as a set of smooth parametric curves. The physical instantiation of the parametric curve is a set of pixel points. There is however the possibility to estimate the analytical definition of the curves from the digital picture. There is also another possibility of approximation: A discrete curve is approximated by the polygonal line segments of neighboring pixels of the trace of the curve. This technique will be used to implement the discrete UNL transform.

In order to be able to estimate analytical equations from discrete pixels another condition must be satisfied: The discrete line patterns must have a thickness of only one pixel. This makes the curves traceable for algorithms that follow the curves.

5.2.2 Theory

The calculus of the UNL Fourier Features is realized in two passes. The pattern composed of a finite set of parametric curves is first transformed from the Cartesian coordinate system to the polar coordinate system by the UNL transform. It is important to notice that this coordinate transformation is performed on entire parametric curves, and not on particular coordinates. The effect of this normalizing transformation is that translations and scale changes of the original pattern are eliminated. Rotations of the original pattern result in cyclic shifts of the UNL transformed pattern. Therefor a second processing pass is applied to the transformed pattern, namely a 2-D Fourier transform. This Fourier transform takes the image (trace) of the UNL transformed curve pattern as input. The spectrum of the Fourier transform is insensitive to any cyclic shifts of its input pattern. Hence also the rotation of the original pattern is eliminated. Characteristics are found which are invariant to translations, scale changes and rotations of the original pattern, the UNL Fourier Features are defined. Figure 5.7 shows the two passes that are necessary in order to calculate the UNL Fourier Features.

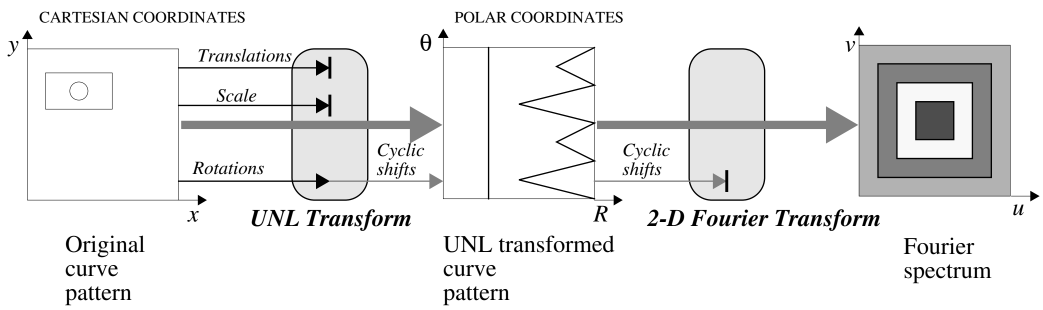


Figure 5.7 Calculus of the UNL Fourier Features.

5.2.2.1 Pattern Synthesis

The general idea to define a pattern of arbitrary complexity is to compose it by a finite number of smooth parametric curves. This technique allows the analytic definition of shapes of an unlimited domain. Open curves can be used to compose a pattern, as well as closed curves. Shapes can be as simple as circles or rectangles, but they can also be nested. Objects can be composed of curves that are detached from each other or that touch other curves.

The formal definition of an object that is composed of smooth parametric curves is

Definition 5.3 **Object composed of smooth parametric curves in 2-D**

An *object* $\Omega(t)$ is composed of the union of n smooth parametric curves $z_i(t)$:

$$\Omega(t) = \bigcup_{i=1}^{n} z_i(t) \qquad z_i(t) = (x_i(t), y_i(t)) \qquad t \in (0, 1)$$

In order to gain an idea of the potential of this pattern definition philosophy, regard Figure 5.8. Here the analytic definition of test pattern from Figure 5.2 is given in terms of Definition 5.3. The origin of the Cartesian coordinate system is placed in the lower left corner of the pattern. Fifteen curves are needed to assemble the final pattern.

$$\begin{array}{cl}
Robot(t) = & \\
\{ (t, 0), (t, 1), (0, t), (1, t), & \textit{Head} \\
(0.2t + 0.2, 0.4(t - 0.5)^2 + 0.6), (0.2t + 0.2, -0.4(t - 0.5)^2 + 0.8), & \textit{Left eye} \\
(0.2t + 0.6, 0.4(t - 0.5)^2 + 0.6), (0.2t + 0.6, -0.4(t - 0.5)^2 + 0.8), & \textit{Right eye} \\
(0.5t + 0.25, 0.5(t - 0.5)^2 + 0.2), (0.5t + 0.25, 0.5(0.6(t - 0.5))^2), & \textit{Lips} \\
(0.2t + 0.4, 0.4), (0.1t + 0.4, 0.2t + 0.4), (0.1t + 0.5, -0.2t + 0.6), & \textit{Nose} \\
(0.1t + 0.2, -0.4t + 1.4), (0.1t + 0.7, 0.4t + 1.0) \} & \textit{Antenna}
\end{array}$$

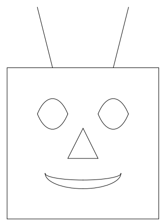
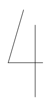


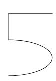
Figure 5.8 Composition of a pattern by a finite set of smooth parametric curves

In Figure 5.9 another kind of pattern is used to demonstrate the pattern synthesis by a set of parametric curves. Characters, like the three digits can also be defined straightforwardly in the sense of Definition 5.3.

$$\begin{array}{cl}
Pattern3(t) = \{ (-2(t - 0.5)^2, 0.5t), & \textit{upper arc} \\
(-2(t - 0.5)^2, -0.5t) \} & \textit{lower arc}
\end{array}$$



$$\begin{array}{cl}
Pattern4(t) = \{ (0.1t + 0.5, 0.3t + 0.3), & \textit{left stroke} \\
(0.2t + 0.5, 0.3), & \textit{horizontal stroke} \\
(0.65, 0.4t + 0.1) \} & \textit{vertical stroke}
\end{array}$$



$$\begin{array}{cl}
Pattern5(t) = \{ (-2(t - 0.5)^2 + 0.8, 0.4t + 0.2), & \textit{arc stroke} \\
(0.5t + 0.3, 0.9), & \textit{horizontal stroke} \\
(0.3, 0.3t + 0.6) \} & \textit{vertical stroke}
\end{array}$$

Figure 5.9 Digits composed of parametric curves

5.2.2.2 Normalization of Translations

The approach of defining patterns as compositions of curves allows the definition of a common centroid of all primitive curves that assemble the complete pattern (EQ 5.12). The relative position of the centroid to the curves of the pattern does not change if the pattern is translated inside the image scene (i. e. the x-y-plane). Therefor a representation of the pattern in polar coordinates with the centroid as origin is insensitive to translations of the original pattern.

Definition 5.4 **Centroid of an object $\Omega(t)$ composed of smooth parametric curves**

$O = (O_x, O_y)$ is the centroid of the object $\Omega(t)$ using EQ 5.12.

5.2.2.3 Normalization of Scale Changes

The idea to compensate for any change of the size of the object, is to express the dimensions of the object in *relative* rather than in *absolute* terms. From the centroid of the object, the distance to a certain pattern point $z_i(t_j)$ will be determined in relation to the maximum distance that occurs. The point of the pattern that is the most distant from the centroid is attributed with a distance of 100%. All other points of the pattern have then a distance to the centroid between 0% and 100%, see Figure 5.10.

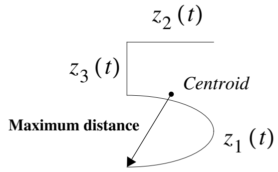


Figure 5.10 Definition of the maximum distance

The distance measure used here is the Euclidean norm. The distance between two points $z_1 = (x_1, y_1)$ and $z_2 = (x_2, y_2)$ is then $d(z_1, z_2) = \|z_1 - z_2\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Definition 5.5 **Maximum distance**
M is the maximum Euclidean distance from O to all curve points:

$$M = max\{\|z_i(t) - O\|\} \qquad \forall i = 1 \dots n \qquad t \in (0, 1)$$

5.2.2.4 Mapping Cartesian Coordinates — Polar Coordinates

The last obstacle on the way to invariant features is the rotation of the curve pattern. Simple template matching algorithms fail if an object is allowed to appear in different orientations.

Recalling the property of the Fourier transform that the Fourier spectrum is insensitive to translations of the original pattern, the approach of the UNL transform is to change rotations of the original patterns in the Cartesian coordinate system to translations (cyclic shifts) in the polar coordinate system.

Therefor the object $\Omega(t)$ is represented in polar coordinates. The UNL transform U of the whole object is defined as the transformation of each individual curve $z_i(t)$ that composes the pattern object:

$$U(\Omega(t)) = U\left(\bigcup_{i=1}^{n} z_i(t)\right) \equiv \bigcup_{i=1}^{n} U(z_i(t))$$

We are now able to define the analytical UNL transform:

Definition 5.6 **UNL transform**
Let $\Omega(t)$ be an object composed of n smooth parametric curves (Definition 5.3). Let O be the centroid of the object (Definition 5.4) and let M be the maximum Euclidean distance from the centroid to all points of all pattern curves (Definition 5.5).
The normalizing mapping of the object $U(\Omega(t))$, respectively the curves that compose it $U(z(t))$ from the Cartesian to the polar coordinate system, using the following formula is the *UNL transform* of the object:

$$U: ((0,1) \to \boldsymbol{R}^2) \to ((0,1) \to \boldsymbol{R}^2)$$

$$U(z_i(t)) = (R_i(t), \theta_i(t)) = \left(\frac{\|z_i(t) - O\|}{M}, \operatorname{atan}\left(\frac{y_i(t) - O_y}{x_i(t) - O_x}\right)\right)$$

Important note: A curve $z(t) = (x(t), y(t))$ is transformed to a new curve $\varsigma(t) = (R(t), \theta(t))$. The UNL transform maps entire curves U: $(Curve \rightarrow Curve)$ from the Cartesian to the polar coordinate system, and not individual points $(x_i(t_j), y_i(t_j))$ for a particular parameter value t_j.

Definition 5.7 **UNL Fourier Features**

Let $\Omega(t)$ be an object composed of n smooth parametric curves (Definition 5.3). Let $U(\Omega(t))$ be the UNL transform of the object (Definition 5.6) and let $I(U(\Omega(t)))$ be the image (trace) of $U(\Omega(t))$ (Definition 5.2), i.e. all points of which the UNL transformed curves are composed of (the materialization of the UNL transformed pattern).

The Fourier spectrum $\| \mathcal{F}\{I(U(\Omega(t)))\} \|$ of the image $I(U(\Omega(t)))$ of the UNL transform $U(\Omega(t))$ of the object $\Omega(t)$ are the *UNL Fourier Features* of the object.

Lemma 5.3 (proof in Annex A1). **Invariance of the UNL transform**

The UNL transform $U(\Omega(t))$ of the object $\Omega(t)$ is invariant to translations of $\Omega(t)$ by the offset Δz and invariant to scale changes by a scalar a. A rotation about any point $z_r = (x_r, y_r)$ by the angle $\Delta\theta$ has no effect on the first variable $R(t)$ of the UNL transform and causes a 2π-periodic translation of the second variable $\theta(t)$ by $\Delta\theta$ (cyclic shift).

Theorem 3.3 **Invariance of the UNL Fourier Features**

The UNL Fourier Features are invariant to translations, scale changes and rotations of the original object $\Omega(t)$.

Proof: This is true following the translation property of the Fourier transform (Lemma 5.2) and Lemma 5.3.

5.2.2.5 Examples of the Analytical UNL transform

In order to illustrate the effects of the UNL transform, some examples are presented. The first shape is shown in Figure 5.11. On the left, a rectangular box with the dimensions 2×1. Inside the box, a circle with radius $1/4$ is located. The center of the circle matches with the center of the box. The whole object is made up of 5 smooth parametric curves, the four lines that limit the box and the circle. The centroid of the whole object coincides with the center of the circle. The maximum distance $M = \sqrt{5}/2$ exists four times at the corners of the rectangle at 0.46, 2.67, -0.46 and -2.67. On the right side, the UNL transform of the pattern is shown. The circle is transformed into a straight line, since the distance $R(t)$ is constant for all angles $\theta(t)$. The four lines that define the rectangle are transformed into "V"-shaped forms. The two horizontal lines of the rectangle become the large "V"s, and the vertical lines of the rectangle become the small "V"s. The "V" of the left vertical line of the rectangle is split into two parts, one extending from 2.67 to π and the other from -2.67 to $-\pi$.

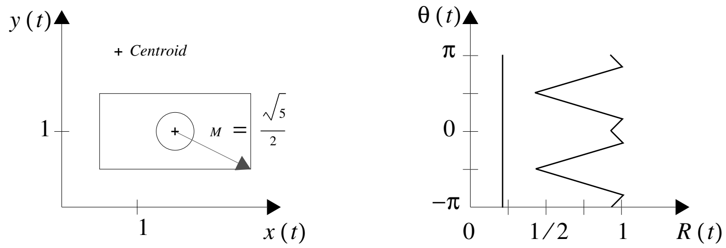


Figure 5.11 Analytic UNL transform of a simple shape

The next examples are the digits defined in Figure 5.9. In the lower row of Figure 5.12 the UNL transform of the patterns of the upper row is shown. The centroids of the individual digits is marked with a cross. Corresponding curve coordinates are labeled with the same letter, both in the Cartesian coordinate system (upper row) and in the polar coordinate system (lower row).

Consider for instance the coordinate at label d in the pattern for the digit "4". It has an angle of

$\theta(t_j) = \pi/3 = 60°$ relative to the horizontal reference axis. For the parameter $t_j = 1$ is valid, since d is situated at the end of the curve. The Euclidean distance from the coordinate at label d to the centroid O is about 60% compared to 100% of the maximum distance M between the coordinate at label e and O. It is noteworthy that the “3” and “4” pattern are of quite different size, but appear normalized in the polar coordinate system.

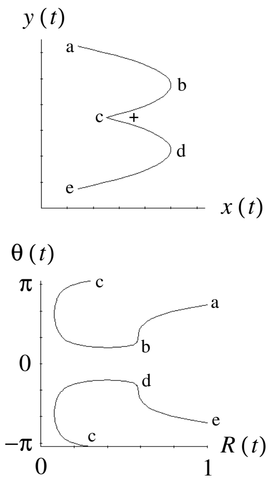


Figure 5.12 Analytic UNL transform of digits

The last example is the test pattern of Figure 5.2 respectively Figure 5.8. In Figure 5.13 the analytic UNL transform of the test pattern is displayed. Also here, corresponding coordinates are marked with the same label in both Cartesian and polar coordinates. Consider the point at label f. The vector from the centroid O has an angle of $\theta(t_j) = \pi/4 = 45°$ relative to the horizontal reference axis and has a distance of about 75% in relation to the maximum distances at point a or b.

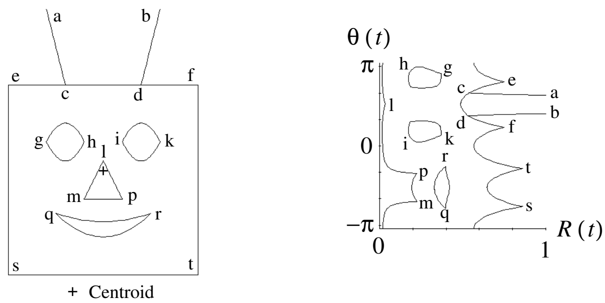


Figure 5.13 Analytic UNL transform of the test pattern

The UNL transform of the digits and the toy robot head were analytically generated by means of the *Mathematica* software package [Wolfram, 1991].

5.2.3 Calculus of the Discrete UNL Fourier Features

The concepts presented so far aim to describe two-dimensional visual patterns. The theory of the UNL Fourier Features would not offer a practical method to classify shapes, if the theoretical concepts were not feasible. In this section, the particular problems of the practical implementation of the calculus of the UNL Fourier Features are outlined. Proposals will be made to solve these problems in order to obtain a robust, multi-purpose shape descriptor in the case when the pattern is a digital binary image.

The discrete UNL Fourier Features are calculated also in a two step procedure. First the UNL transform of the discrete pattern is calculated. Then the discrete 2-D Fourier transform (DFT) is applied to the UNL transformed pattern. This yields the final UNL Fourier Features in the discrete case. It is however important to mention that the UNL transform is always done analytically. From the discrete pattern the analytical curve equations are estimated, the analytical UNL transform is performed, and finally the transformed curves are instantiated in the polar coordinate system.

5.2.3.1 Discrete Pattern

In practice a visual 2-D binary curve pattern is given as a binary bitmap. Some of the pixels of the image are set (where there is a point of a curve) and the majority of the pixels of the image is blank. In Figure 5.14 an example of a binary discrete image is shown. The dark pixels belong to the object, the white pixels belong to the background. The resolution of the square image is 8×8. The low resolution is chosen for illustrative reasons only.

The first fundamental problem of a discrete pattern is:

Where are the analytical definitions of the parametric curves that compose the pattern?

Unless in the case that the patterns were defined in an analytical form, for instance in a CAD system, the analytical definition of the curves is not available. Only a rectangular grid of black and white pixels can be used to generate the desired shape features.

The general idea to obtain an analytical definition of curves from a discrete bitmap is to *approximate* and *estimate* the pattern curves by line segments of neighboring pixels. In a binary image it is difficult to determine where one curve stops and the other begins. Often this kind of strategy involves the use of heuristics which limit the general purpose character of a certain method.

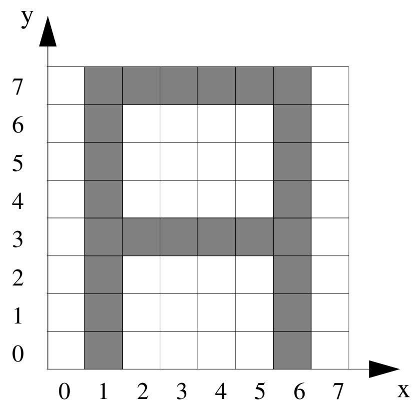


Figure 5.14 A discrete binary pattern

5.2.3.2 Pattern Preprocessing

The most important condition of an input pattern for the discrete UNL transform is that the pattern curves are thin. That means that it must be possible to determine pixels that belong to the pattern defining curves, and not to some interior region of the pattern.

A *thinning* or *skeletonizing* preprocessing step must therefor be applied to the image first. From a “fat” pattern which has pattern parts larger than a one-pixel thickness, a “thin” pattern or skeleton is obtained. Normally these thinning algorithms are iterative local operators that pass over the image several times, until the desired thin pattern is obtained. In all experiments presented in this thesis, the algorithm of Zhang and Suen was used [Zhang, 1984], see also [Gonzalez, 1987].

Besides the thinning, a further preprocessing step was applied. In order to ensure that there are not ambiguities how to follow the curve, the condition of *m-connectivity* [Gonzalez, 1987] was required for the curve patterns. In the example pattern of Figure 5.14 there are two possible paths to follow from pixel $(1, 6)$ to pixel $(2, 7)$, directly or via the pixel $(1, 7)$. Since we want to approximate the curve by line segments, it is desirable that the same piece of curve is not represented by two different line segments. In order to achieve the condition of m-connectivity, a 3×3 mask is repeatedly passed over the pattern which searches for particular pixel constellations and eventually deletes the central pixel of the mask. This is done until no more changes occur between two consecutive passes.

The thinning and m-connectivity preprocessing is first performed with the original pattern in the Cartesian coordinates. After the discrete UNL transform has been performed, the transformed pattern is represented in the polar coordinate system. Also here, the two preprocessing steps are applied. This is justified by the fact that we want to obtain one-pixel thick curves in the polar coordinate system. The UNL transform of the original pattern eventually causes situations where the pattern in the polar coordinates has parts with a thickness of more than one pixel.

5.2.3.3 Line Segment Approximations of Discrete Object

The simplest approach to approximate a curve pattern in the discrete case is to compose it by all line segments of neighboring pixels. Two neighboring pixels $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$ of the curve pattern define a line segment, with $|x_i - x_j| \le 1$ and $|y_i - y_j| \le 1$. The basic assumption that is made, is that all these line segments between neighboring pixels approximate the analytical curves sufficiently well. Theoretically, for an infinitely high resolution, the line segments become points of the curve.

One advantage of this approximation technique is that it is irrelevant to define a discrete curve pattern by a finite set of continuous curves. The property of continuity of the curve becomes more difficult to define, the lower the resolution of the image becomes. In order to segment the whole pattern into continuous parts, it would be necessary to search for the break points where two different continuous curves meet. The composition of the whole pattern by tiny line segments liberates from this complex search procedure.

The most important statement at this point is that there is an *analytical* definition from a *discrete* line segment between two points $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$:

> The analytic equation of the line segment between the two discrete coordinate points $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$ is $z_{ij}(t) = (x_i + t(x_j - x_i), y_i + t(y_j - y_i)) \qquad t \in (0, 1)$

This approach permits to define the discrete version of the analytical definition of an object:

Definition 5.8 **Discrete object in 2-D**

A *discrete object* $\hat{\Omega}(t)$ of a pattern consisting of n pixels $z_i = (x_i, y_i)$ is composed of the union of m line segments $z_{ij}(t)$ between all neighboring pixels $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$ of the pattern:

$$\hat{\Omega}(t) = \bigcup_{i=1}^{m} z_{ij}(t) \qquad z_{ij}(t) = (x_i + t(x_j - x_i), y_i + t(y_j - y_i)) \qquad t \in (0, 1)$$

The number m of line segments is approximately the number of pixels n and the limiting value of m is n, if the resolution is infinitely high. For instance if the pattern is consisting of an open, noncrossing, thin and m-connected curve of n pixels, the number of line segments will be $n - 1$.

5.2.3.4 Discrete Centroid and Maximum Distance

The next problem to achieve the UNL transform in the discrete case is the definition of the discrete centroid $\hat{O}$. It is possible to use the analytical definition of the centroid (EQ 5.12). By simple algebra one can show that $\hat{O} = \frac{1}{m}\sum_{i=1}^{m}\frac{z_j - z_i}{2}$. The higher the resolution, the more does the value $\frac{z_j - z_i}{2}$ approximate the value of z_i. Hence the centroid of the discrete curve pattern can be defined as the mean (expected) value of the m pixels that compose the pattern:

Centroid of the discrete pattern: $$\hat{O} = E\{z_i\} = \frac{1}{m}\sum_{i=1}^{m} z_i$$

The maximum distance $\hat{M}$ is simply the maximum Euclidean distance of the pixels that compose the pattern to the discrete centroid $\hat{O}$.

$$\hat{M} = max\{\|z_i - \hat{O}\|\} \qquad \forall i = 1 \ldots m$$

5.2.3.5 UNL transform for the Discrete Pattern

With these propositions, we are able to define the UNL transform of the discrete pattern. Once again, the UNL transform of the whole object is defined as the union of the UNL transform of all line segments that compose the pattern:

Definition 5.9 **UNL transform for discrete pattern**

Let $\hat{\Omega}(t)$ be a discrete object composed of n pixels $z_i = (x_i, y_i)$. Let $\hat{O} = (\hat{O}_x, \hat{O}_y)$ be the centroid of the object and let $\hat{M}$ be the maximum Euclidean distance from the centroid $\hat{O}$ to all pixels z_i.

The normalizing mapping of the object $U(\hat{\Omega}(t))$, respectively the line segments $z_{ij}(t)$ between two neighboring pixels $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$ that compose the object, $U(z_{ij}(t))$ from the Cartesian to the polar coordinate system, using the following formula is the *UNL transform* of the discrete object:

$$U: ((0,1) \to \boldsymbol{R}^2) \to ((0,1) \to \boldsymbol{R}^2) \qquad \forall i = 1 \ldots n \qquad t \in [0,1]$$

$$U(z_{ij}(t)) = \varsigma_{ij}(t) = (R_{ij}(t), \theta_{ij}(t)) = \left(\frac{\left\| z_i + t(z_j - z_i) - \hat{O} \right\|}{\hat{M}}, \operatorname{atan}\left(\frac{y_i + t(y_j - y_i) - \hat{O}_y}{x_i + t(x_j - x_i) - \hat{O}_x} \right) \right)$$

After the line segments $z_{ij}(t)$ have been transformed to curves $\varsigma_{ij}(t)$ *analytically*, they must be *instantiated*. That means that a discrete curve pattern must be generated in the discrete polar coordinate system. This is easily achieved by quantizing the continuous parameter t into k discrete steps. The only condition for the generation process in the polar coordinate system is that no gaps may occur when the parameter t is gradually increased from 0 to 1 in k steps. Hence k must be sufficiently large. The size of k is determined by the resolution of the image in the polar coordinate system. In the current implementation, k is calculated by an experimentally defined heuristic: $k = 1 + ROUND(2\pi \| \varsigma_{ij}(1) - \varsigma_{ij}(0) \|)$. The following piece of code realizes the instantiation of an analytical curve $\varsigma_{ij}(t)$ in the polar coordinate system.

Algorithm 5.1 Instantiation of the UNL transform of a line segment

```
Let zi=(xi,yi) and zj=(xj,yj) be two neighboring pixels of the pattern
Let k be the number of the discretization steps
Let O be the centroid of all pixels zi
Let M be the maximum Euclidean distance from O to all zi
Let N be the resolution of the polar coordinate system of a (N x N) image

Algorithm: Discrete UNL transform of the line segment between zi and zj

1.)x ← xi; y ← yi; dx ← xj-xi; dy ← yj-yi
2.)For i=0...k:
      t ← i/k
      x ← xi + t * dx; y ← yi + t * dy
      vx ← x - Ox; vy ← y - Oy
      R ← sqrt((vx)^2 + (vy)^2)
      θ ← atan2(vy, vx)
      pR ← ROUND(N * R/M)
      if(θ >= 0)
          pθ ← ROUND(N * θ/(2π))
      else
          pθ ← ROUND(N * (θ+2π)/(2π))
      Switch the pixel (pR,pθ) in the polar coordinate system on
```

The function "ROUND" returns the rounded integer value of its argument.
The function "sqrt(x)" calculates the square root of its argument.
The function "atan2(y, x)" returns the principal value of the arc tangent of y/x, using the signs of both parameters to determine the quadrant of the return value. The return values are in the range [-π, π].

In Figure 5.15 the UNL transform of a discrete pattern is resumed. The thinned and m-connected pattern in Cartesian coordinates is scanned for neighboring pixels (One pass over the whole image is sufficient to detect neighbors). Two neighbors $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$ define then the analytical line segment $z_{ij}(t)$. By use of Algorithm 5.1, the analytical UNL transform $\varsigma_{ij}(t)$ of the line $z_{ij}(t)$ is instantiated. The pattern in polar coordinates is thinned and m-connected once again to ensure one-pixel thickness.

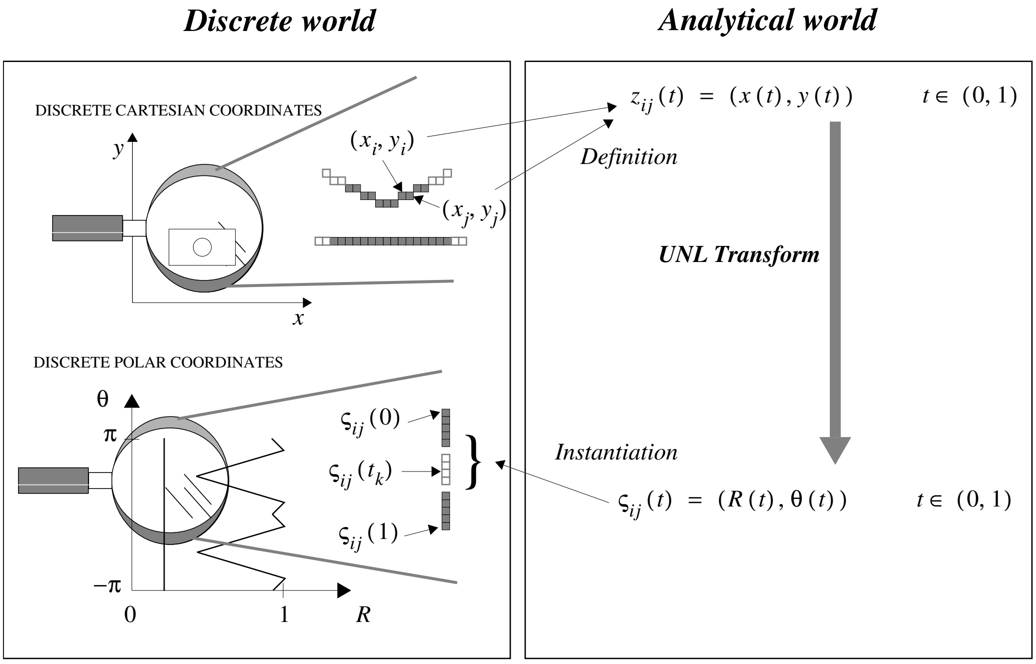


Figure 5.15 Discrete UNL transform

5.2.3.6 Discrete Fourier transform

Finally the UNL Fourier Features $F(u, v)$ of the discrete pattern can be calculated. The discrete two-dimensional Fourier transform (EQ 5.10) is applied to the UNL transformed pattern.

Some interesting properties of the 2-D Fourier transform must be taken into consideration to obtain the discrete UNL Fourier Features.

A.) *Conjugate symmetry*

Since the input for the Fourier transform is a 2-D binary image, consisting of a real, discrete function with values 0.0 and 1.0, the Fourier spectrum exhibits the property of conjugate symmetry: $|F(u, v)| = |F(-u, -v)|$ ([Gonzalez, 1987]). This signifies that half of the features are identical and therefor redundant.

B.) *Signal normalization*

The magnitude $F(0, 0)$ represents the basic signal energy of the pattern. Furthermore it is proportional to the *average value* $\bar{f}(x, y)$ of the 2-D discrete function $f(x, y)$. We suppose that identical patterns will also have identical average values. Therefor all magnitudes of the Fourier spectrum will be normalized by dividing them by $F(0, 0)$. Since $F(0, 0)$ is always the maximum of all $F(u, v)$, this normalization has the desirable effect that all UNL Fourier Features are scaled to the interval $[0, 1]$.

The definition of the 2-D shape descriptor from a real world image can finally be given:

Definition 5.10 **UNL Fourier Features of a discrete pattern**

Let $i(x, y)$ be a two-dimensional image that represents a discrete object $\hat{\Omega}(t)$ and let $f(R, \theta)$ be the a two-dimensional image that represents the UNL transform of $\hat{\Omega}(t)$. The normalized discrete Fourier spectrum $UFF(u', v') \equiv \| \mathcal{F}\{f(R, \theta)\} \| / F(0, 0) = \| F(u, v) \| / F(0, 0)$ the of the image $f(R, \theta)$, ignoring $F(0, 0)$ and the values that are duplicated by conjugate symmetry are the discrete *UNL Fourier Features* of the object $\hat{\Omega}(t)$.

Hence the number of UNL Fourier Features depends only on the resolution N of the polar coordinate system. Typically values for N are 2^n, where n is a natural number. A resolution of N yields N^2 magnitudes. After subtracting the duplicated values caused by the conjugate symmetry, $N^2/2 + 2$ values remain. $UFF(0, 0)$ has always the value of 1.0 and it therefor worthless for classification. Consequently the number of UNL Fourier Features is $N^2/2 + 1$ for resolution N.

For the discrete calculus, the indices u' and v' of $UFF(u', v')$ must obey the conditions:

For all (u', v') with $0 \le u' \le N - 1$ and $0 \le v' \le N - 1$ it is

i.) $(u', v') \neq (0, 0)$

ii.) $v' \le N/2$

iii.) if $v' = 0$ or $v' = N/2$ then $u' \le N/2$

Note: It verifies that the most discriminant UNL Fourier Features $UFF(u, v)$ have low values for the v-parameter and have values for the u-parameter that are either near 0 or near N. Under these circumstances it is convenient to take advantage of the periodicity of the u-parameter and express the indices close to N as $u{-}N$. This index transform ensures that only small absolute values for the indices occur. For instance with $N = 128$, $UFF(127, 2)$ then becomes $UFF(-1, 2)$.

Furthermore the fact that only low frequencies are discriminative in practice, allows the a priori reduction of the UNL Fourier Features by bounding the range to the low frequencies. A fraction q can be introduced which limits the indices u' and v' even further.

iv.) $v' \le (N/2)/q$

v.) if $u' \ge N/2$ then $N - u' \le N/q$ else $u' \le N/q$

For instance with a fraction $q = 8$ and a resolution of $N = 128$ for the discrete polar coordinate system, we obtain the reduced set of 280 features.

5.2.3.7 Computational Complexity

The calculus of the UNL Fourier Features is not expensive. Starting from a binary image, the first step is the thinning and m-connectivity of the pattern. These steps are eventually time consuming, especially when the pattern’s interior region is relatively large. Then a single pass over the image is done to detect all neighboring pixels. For each of these neighbors, Algorithm 5.1 must be called. The number of operations in the algorithm is proportional to the number k of quantization steps. Once again a thinning and m-connectivity step is applied to the transformed pattern. Then a 2-D Fourier transform is performed.

A classifier that is based on the UNL Fourier Features, calculates *all* features in its training phase, using a Fast Fourier transform (FFT). After the most discriminating features have been determined by a feature selection algorithm, only a few UNL Fourier Features must be calculated in the time critical classification phase. This can done explicitly using the formula of EQ 5.10:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)\, e^{\frac{-j2\pi(ux + vy)}{N}}$$

Besides the fact that only a few indices u and v must be used, the majority of the $f(R, \theta)$ is zero, since only a few pixels of the image are part of a pattern curve. This liberates from the calculus of the complex exponential part of the term on the right side of the summation. The following routine calculates one UNL Fourier Feature in practice.

Algorithm 5.2 Calculus of a single UNL Fourier Feature

```
Let Np be the resolution of the polar coordinate system.
Let P={f(R,t) | f(R,t)=1, 0<=R<Np, 0<=t<Np} be the image of the UNL
   transformed pattern of the original pattern.
Let C be the cardinality of P (=Number of object pixels).
Let SinTab be a lookup table of dimension Np that contains the precalcu-
   lated value for the index i, 0<=i<Np: sin(2πi/Np)
Let CosTab be a lookup table of dimension Np that contains the precalcu-
   lated value for the index i, 0<=i<Np: cos(2πi/Np)

Algorithm: Discrete UNL Fourier Feature: UFF(u,v)

1.)Re ← 0; Im ← 0;
2.)For j=1...C:
     (Rj,tj)are the coordinates of the next point of P.
     Calculate the index of the lookup table: i ← ((u*Rj+v*tj) mod Np);
     Re ← Re + CosTab[i]; Im ← Im + SinTab[i];
3.)Re ← Re / Np; Im ← Im / Np;
4.)UFF(u,v)← sqrt(Re*Re+Im*Im); (Euclidean norm)
```

5.2.4 Comparison of the Log-Polar transform and the UNL transform

The differences between the UNL transform (*UNL*) and the Log-Polar transform (Complex-Log transform) (*CL*) must be stated explicitly because both techniques have some concepts in common.

Both seek a translation normalization by shifting the centroid of the object pattern to the origin of a polar coordinate system. Both map from a pattern, originally in Cartesian coordinates to polar coordinates. The fundamental differences however are:

i.) *Application Constraints:*
While *UNL* can be applied to every pattern composed of curves, *CL* is limited to patterns which posses an interior region. For the test pattern of Figure 5.2, *CL* is not suited, e.g. it could not meaningfully be used to transform the antenna.

ii.) *Subject of Mapping:*
CL maps image points through a mathematical function to other image points. A complex coordinate z is transformed to $ln(z)$. The axes of the original x-y-pattern change to $(ln\|z\|, \theta)$. *UNL* is “just” a coordinate transform. It does not change the value of z but rather represents it in another coordinate system.

iii.) *Mathematical Operations:*
CL calculates $ln(z)$, which yields another complex number. *UNL* is mathematically an identity operation. The value of z is invariant.

iv.) *Scale invariance:*
UNL is analytically scale invariant for every positive scale factor. Limitations occur only in the discrete pixel matrix when the differences in the scale factor become too large. It was already outlined in section 5.1.2.2 that *CL* is only obviously scale invariant. The problem of *stretching* was mentioned.

5.2.5 Limitations

The application domain of the UNL transform are curve-like, binary patterns. This constraint excludes a large amount of pattern recognition tasks based on artificial vision. It is not possible to perform classification tasks that take a different kind of visual information into consideration. For instance if the color or the texture of an object are indispensable for recognition, the method presented here is surely inappropriate. Nevertheless, the spectrum of 2-D patterns that can be assembled by thin curves is unlimited which confirms the general purpose applicability of the UNL Fourier Features as a 2-D form descriptor.

The use of the UNL Fourier Features as a two-dimensional shape descriptor assumes an ideal segmentation if the source images are non-binary, but so do other well established methods as well, for instance 1-D Fourier descriptors. On the other hand, patterns are often a priori only black and white, e.g. printed digits. In this case, no segmentation is necessary.

The UNL Fourier Features are sensitive to occlusions. The set of parametric functions is changed abruptly if parts of the whole object are missing. Consequently the centroid and the UNL transform of the original pattern change completely and the UNL Fourier Features yield garbage values.

5.2.6 Experiments

In the following section, practical experiments are carried out that corroborate the usefulness of the UNL Fourier Features as a general purpose 2-D shape descriptor. The features are used as a base for a classifier in the sense of [Devijver, 1982].

5.2.6.1 Benchmark Patterns

Two different pattern families are used for the classification experiments.

The first type of patterns are the results of a segmentation of 2-D shapes using the image processing techniques described in [Gonzalez, 1987]. An image is captured by a standard CCD camera. The digitalized grey image is processed by the vision system until a binary image is obtained. From the binary image, the boundaries of the object itself and of the nested objects inside the object (=holes) are extracted. The camera position was varied ten times along the surface normal of the working table from an initial distance of $0.6m$ with $1cm$ increases. This resulted in a scaling of about 115% from the smallest to the largest sample of each class. For each object class there are about 150 training samples and 50 test samples. After the segmentation, the horizontal chord segments that compose the object were multiplied by a constant linear scale factor of 1.088 in order to try to compensate for the distortion of the CCD camera. The system calls of the vision system were used to extract the boundaries of the objects respectively of their holes. The whole set of images for one class was stored in a file and then transferred to a workstation for further processing. The resolution of the original image will be denoted as N_C. In the case of the Magiscan-2 image processing system, we have $N_C = 512$. In Figure 5.16, thirteen different 2-D curve patterns are displayed that were obtained by image processing techniques.The original objects are partially 3-D objects, like object 1 and object 3 and partially flat paper sheets, like object 5 or object 13. Together with the thinned and m-connected original curve pattern on the left column, the image of the UNL transform of the respective pattern is shown in Figure 5.16. The first training pattern of each class is displayed. For the generation of the figure, the resolution of the Cartesian coordinate system was $N_C = 512$ and for the polar coordinate system $N_p = 512$.

All patterns are closed curves. Only the objects 4, 9 and 13 contain nested curves. These three objects have an identical outside border and two holes. Each of the holes can be found in one of the other patterns, e.g. the round hole of pattern 9 is the same as the round hole of pattern 4, and the rectangular hole of pattern 9 is identical to the rectangular hole of pattern 13. Objects 5 and object 6 are symmetrical.
Images that were not properly segmented into binary images were discarded a priori. It does not make sense to feed a 2-D shape descriptor with contours that were obtained from totally degraded binary images. For each of the 2-D shape descriptors presented by now, garbage values would result from a badly segmented image. Therefor it can be justified to use only curves that were obtained from binary images that contain a representative 2-D silhouette of the 3-D object.

The second type of patterns are handwritten digits. These kind of curves pose an especially great challenge to pattern recognition. Besides the “syntactic” noise that an image can contain, for instance illumination changes, a “semantic” noise is contained in the image. A person never reproduces exactly the same pattern twice. The pattern prototype changes over time. Even more difficult is a writer-independent recognition of characters. In the presented experiments only single-writer data sets will be examined.

For the calculus of the UNL Fourier Features, the UNL transformed pattern is generated in the polar coordinate system. The resolution of the polar coordinate system N_P can be different from the resolution of the original pattern in Cartesian coordinates N_C. In all experiments N_P is set to $N_P = 128$. Even beforehand, the original pattern can be scaled down to a lower resolution $N'_C < N_C$ and then the transformed pattern is generated in the polar coordinate system.

5.2.6.2 Feature Selection and Performance Evaluation

In order to compare several shape descriptor methods to each other, the classifier model will always be the same. From the original feature set, a reduced number is selected. Then an error estimation is performed using a Leave-One-Out estimation [Devijver, 1982]. The classifier model will be a Nearest Neighbor with Euclidean distance measure. No data compression of the training set is performed, i.e. the distance to each sample of the training set is measured.

A particular problem surges with the calculus of the UNL Fourier Features. Since the number of the calculated features is very high, even when an a priori reduction has been done, a *feature preselection* will be performed. That means that the original data set is first reduced to an intermediate number of features. The number of preselected features P is fixed to $P = 100$. Then from this intermediate number the final feature set is selected. For the preselection, a suboptimal technique is employed. The search method is the “Best Features” algorithm ([Devijver, 1982]). The selection criterion is the estimated error, using only one feature ($k = 1$). It would be too costly to use the search strategy of “Sequential Forward Search” of [Devijver, 1982]. For the selection of the final feature set, the MINERR method is used (Minimum estimated classification error, using a 1-Nearest Neighbor classifier with Euclidean distance together with a Leave-One-Out estimator). For instance for a resolution $N_P = 128$ of the polar coordinate system, we obtain $N_P^2/2 + 1 = 8193$ UNL Fourier Features. From these amount, the preselection chooses $P = 100$ features. Then the MINERR algorithm is used to determine an error rate in dependence of the number of selected features.

5.2.6.3 Experiment: Comparison of the Fourier Descriptors and Invariant Moments to the UNL Fourier Features for a Benchmark of Typical 2-D Vision Object Recognition

In the first experiment, a direct comparison of the classical shape features Fourier descriptors of 5.1.1.4 and the moments invariants of 5.1.2.1 to the UNL Fourier features is made. Different subsets of the objects of Figure 5.16 are used to extract both Fourier Descriptors, invariant moments and UNL Fourier Features. The best of the features for each descriptors will be used to estimate the error of a classifier which model is described above. For the calculus of the Fourier descriptors, the original pattern with a resolution of $N_C = 512$ is thinned and m-connected. Here, the property of m-connectivity is especially important because the curve must be traced. Only patterns with closed boundaries are used. The boundary is traced in counterclockwise direction and the coordinates of the boundary points are stored in an array A. Then this array is proportionally mapped to a fixed sized buffer B with length $M = 2^{11} = 2048$, using linear interpolation of the coordinates. That means that two neighboring points of the original array A are mapped to two new points of the fixed sized array B. Any gaps that eventually occur between the two new points in B are filled in by linear interpolation. A Fast Fourier transform is applied to the fixed size array B, generating the Fourier transform F. Only the magnitudes of the Fourier transform are used as the shape features. Furthermore indices in the upper half of the index domain, i. e. indices $u \geq M/2$ are expressed as $u' = u - M$. This gives a more convenient representation of the indices with small absolute values. $F(0)$ contains no invariant information and is therefor discarded. $F(1)$ has the largest size for a closed, counterclockwise traced curve [Gonzalez, 1987] and is therefor used as the normalizing factor. F is divided by $F(1)$ which yields the final array FD with the Fourier descriptors. Now $F(1)$ is also useless for classification and discarded. The number of Fourier descriptors of the boundary curve is therefor $M - 2 = 2046$. Also here, the same feature preselection as for the UNL Fourier Features is used, followed by the MINERR selection. The preselection is also necessary because of the large number of features. Finally the error is estimated.

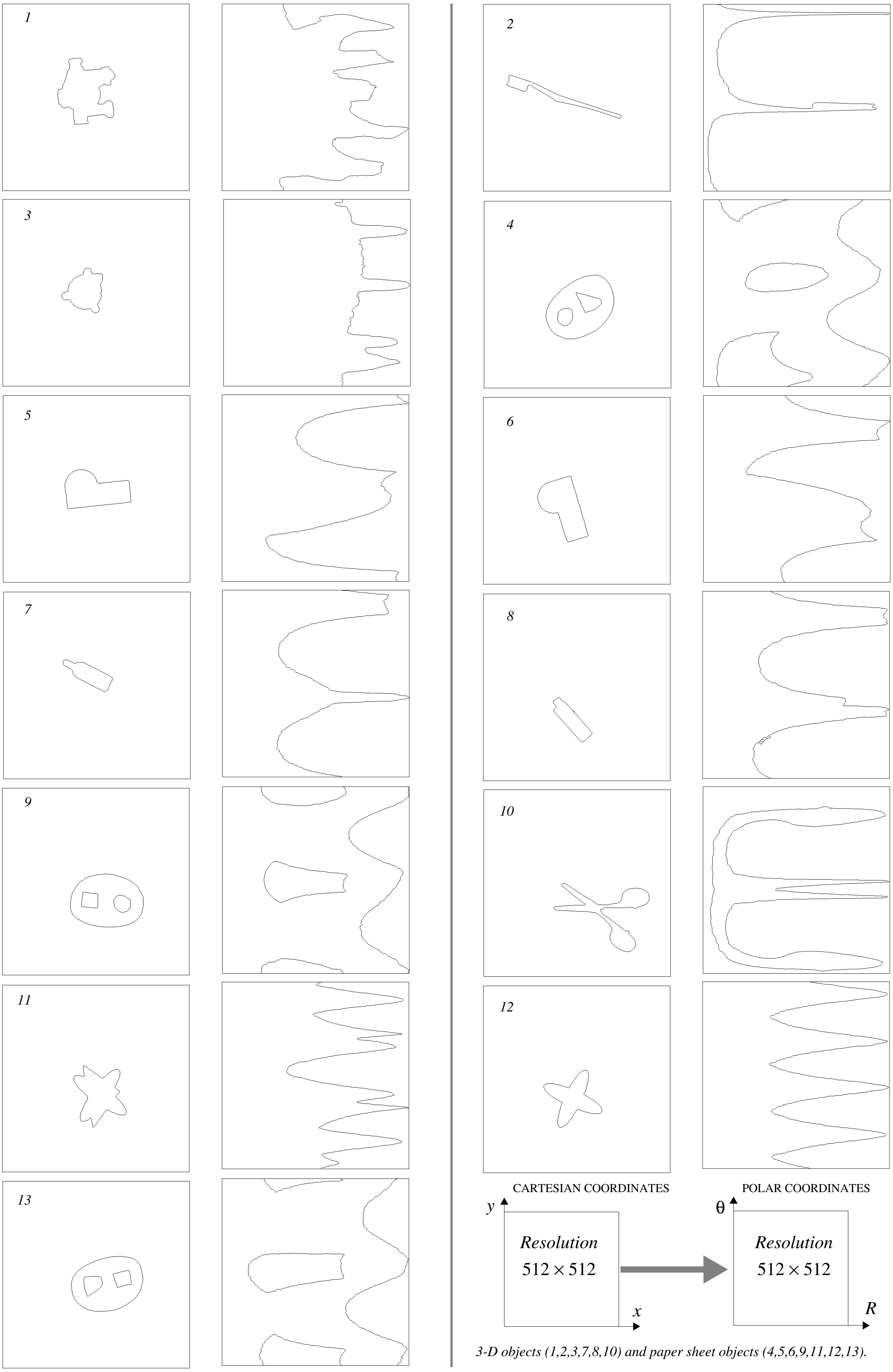


Figure 5.16 Benchmark test set. Original pattern and image of the UNL transform in polar coordinates.

Since the number of the *invariant moments* ϕ_i, $i = 1\ldots7$ is only seven, no feature preselection is necessary. All features are submitted to the MINERR selection and the estimated error is reported. It should be noted that moments are normally applied to patterns with an interior region.

For the calculus of the UNL Fourier Features, the original resolution of the Cartesian coordinate system $N_C = 512$ is reduced to $N'_C = 128$ by a linear down-scaling of the curve. After thinning and m-connectivity, the UNL transform of the discrete pattern is performed, generating the transformed pattern in polar coordinates with a resolution of $N_P = 128$. Once again, thinning and m-connectivity is applied. Finally the discrete 2-D Fourier transform is calculated. Feature preselection, selection and error estimation follow.

Subset 1: The first subset of objects that was used to compare the three shape descriptors was $\{1, 3, 7, 8\}$ from Figure 5.16. The results of the feature selection and error estimation experiments are summarized in Table 5.1 and Figure 5.17.

Table 5.1 Feature selection and error estimation for the object set $\{1,3,7,8\}$

	2-D SHAPE DESCRIPTOR											
	Fourier descriptors $F(u)$				*Invariant moments* ϕ_i				*UNL Fourier Features* $UFF(u, v)$			
	Estimated error [%]											
	Preselection		*Selection*		*Preselection*		*Selection*		*Preselection*		*Selection*	
Feature Nr.	Index u	Criterion	Index u	**Criterion**	i	Criterion	i	**Criterion**	(u, v)	Criterion	(u, v)	**Criterion**
1	3	13.62	3	**13.62**	-	-	3	**17.07**	-1,2	0.52	-1,2	**0.52**
2	-1	14.32	-4	**1.38**	-	-	2	**1.21**	1,0	4.65	1,0	**0**
3	-4	14.31	-1	**0**	-	-	4	**0.34**	-3,3	8.79	-3,3	**0**
4	-2	16.55	-2	**0**	-	-	5	**0.17**	-1,4	10.00	-1,4	**0**
5	-10	22.76	-10	**0**	-	-	1	**0.17**	0,3	10.52	0,3	**0**
6	5	23.28	5	**0**	-	-	7	**0.17**	2,0	15.34	2,0	**0**
7	-7	25.17	-7	**0**	-	-	6	**0.34**	1,2	18.28	1,2	**0**
8	-3	25.69	-3	**0**	-	-	-	-	2,8	20.69	2,8	**0**
9	4	26.21	4	**0**	-	-	-	-	2,10	22.59	2,10	**0**
10	-5	26.38	-5	**0**	-	-	-	-	-1,3	22.76	-1,3	**0**

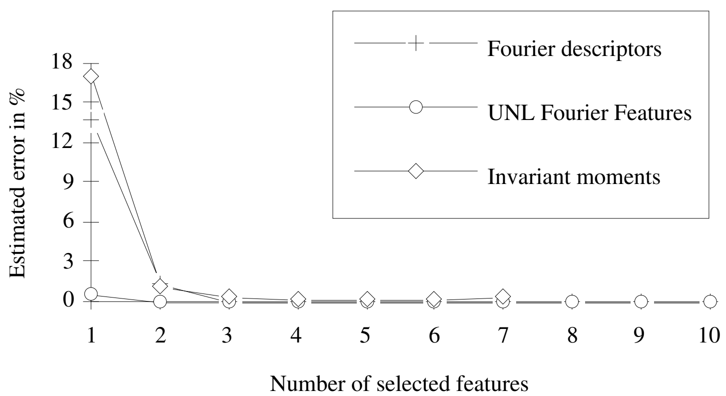


Figure 5.17 Estimated error as a function of the selected features for the object subset $\{1,3,7,8\}$

The table says e.g. that for the Fourier descriptors, in the preselection phase $F(5)$ of the FFT estimated an individual error of 23.28%, using the “Best Features” search in combination with the minimum error selection criterion. Then in the selection phase, using MINERR, for the first two features $F(3)$ and $F(-4)$, an error of 1.38% occurred. For the invariant moments there is of course no preselection, due to the small number of features.

It can be observed that the UNL Fourier Features show the best result. Already with the features set $UFF(-1, 2)$ and $UFF(1, 0)$, a 100% accuracy of the classifier is predicted. The Fourier descriptors converge more slowly to a zero error and the invariant moments never achieve an absolute, error free accuracy.

Subset 2: The next subset of objects $\{5, 6\}$ from Figure 5.16 tests the robustness of the three 2-D shape descriptors against *symmetry*. The two test objects were arbitrarily conceived, with the only condition that there is an axis symmetry. The Fourier descriptors are normally not suited to distinguish symmetric objects.

The setup of the experiment is exactly the same as before, the dimensions of the pattern bitmaps and Fourier buffers, preselection, selection and error estimation are identical.

The graph of Figure 5.18 shows the result of the error estimation. The Fourier descriptors reach a minimum error of 31.33%, using 10 selected out of 100 preselected FDs which corresponds to an accuracy of 68.67%. Note that the theoretical maximum for a randomly determining classifier is 50% for two classes. The conclusion of the error values is that the Fourier descriptors contain almost no discriminatory information. For the invariant moments a maximum accuracy of 83.84% can be achieved, using the first five features $\{\phi_7, \phi_3, \phi_2, \phi_1, \phi_4\}$. Once again the UNL Fourier Features show the best result. The estimated accuracy reaches 100% right from the beginning of the selection and stays there for the next 9 features. The best feature $UFF(1, 9)$ of the preselection is sufficient to separate the two classes linearly.

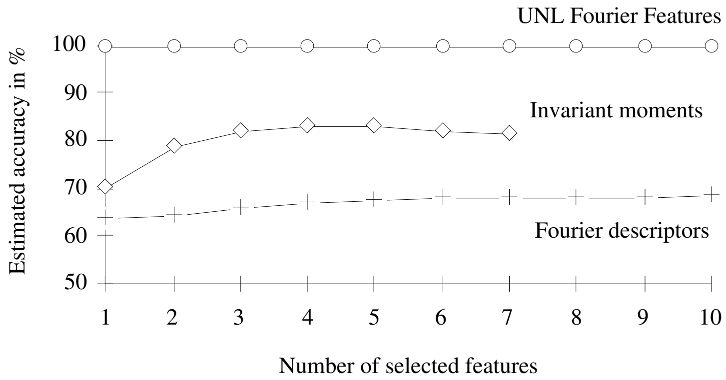


Figure 5.18 Estimated accuracy as a function of the selected features for the object subset {5,6}

At this point, it is once again emphasized that the calculus of one individual UNL Fourier Features is very cheap. In the above example, the first sample of the first pattern (=object 5) has 365 points in the polar coordinate system, before the Fourier transform is applied. For a resolution $N_P = 128$ of the polar coordinate system that means that the remaining $128 \times 128 - 365 = 16019$ pixels are zero ($f(R, \theta) = 0$). Using EQ 5.10 on page 8, only those points of the polar coordinate system must be considered, where the pixel is turned on ($f(R, \theta) = 1$). Furthermore, since the argument of the exponent can only possess N_P different values, lookup tables can be used. This turns the calculus of a single 2-D Fourier magnitude extremely fast.

$$UFF(1, 9) = \frac{1}{128} \sum_{R=0}^{127} \sum_{t=0}^{127} f(x, y)\, e^{\frac{-j2\pi(1R + 9t)}{128}}$$

Subset 3: The objects $\{4, 9, 13\}$ of Figure 5.16 contain nested curves. Since the outside form is identical, the Fourier descriptors will fail as shape features. Only a more sophisticated, dedicated classifier structure could eventually be able to distinguish among the three objects, for instance calculating the FDs of the holes, classify the holes individually and then decide the class.

Therefor only the invariant moments will be compared to the UNL Fourier Features. Table 5.2 shows the results. The lowest error rate that the invariant moments achieve is 8%. On the other hand, the UNL Fourier Feature set $\{UFF(0, 3), UFF(0, 1), UFF(1, 0)\}$ is able to score a 0% estimated error rate. Once again the UFFs prove their high discriminative power and superiority especially over the Fourier descriptors.

Table 5.2 Feature selection and error estimation for the object set {4,9,13}

2-D SHAPE DESCRIPTOR								
	Invariant moments ϕ_i				*UNL Fourier Features* $UFF(u, v)$			
	Estimated error [%]							
	Preselection		*Selection*		*Preselection*		*Selection*	
Feature Nr.	i	Criterion	i	**Criterion**	(u, v)	Criterion	(u, v)	**Criterion**
1	-	-	6	**41.56**	0,3	11.33	0,3	**11.33**
2	-	-	2	**19.34**	3,2	12.22	0,1	**0.89**
3	-	-	7	**12.67**	2,8	13.33	1,0	**0**
4	-	-	3	**10.44**	1,8	14.67	2,8	**0**
5	-	-	4	**8.44**	1,0	19.33	3,2	**0**
6	-	-	5	**8.00**	0,2	20.89	1,8	**0**
7	-	-	1	**9.33**	3,3	21.78	0,2	**0**

Subset 4: Finally an error estimation for *all* objects of the test set is performed. Additionally to the error estimation, classification experiments are made, using data that is independent from the training data. For each class, about 50 test samples were taken under the same circumstances as the training data. The error rate for the test set is recorded in the column "detected". For instance, using the two best invariant moments reaches an error of 12.69% , feeding the independent test samples to the Nearest Neighbor classifier with Euclidean distance. The error estimation and classification results are presented in Table 5.3.

Table 5.3 Feature selection, error estimation and classification for the set of all objects

2-D SHAPE DESCRIPTOR															
	Fourier descriptors $F(u)$					*Invariant moments* ϕ_i					*UNL Fourier Features* $UFF(u, v)$				
	Error [%]														
		Estimated		*Estimated*	*Detected*				*Estimated*	*Detected*		*Estimated*		*Estimated*	*Detected*
	Preselection		*Selection*			*Preselection*		*Selection*			*Preselection*		*Selection*		
Feature Nr.	Index u	Criterion	Index u	**Criterion**		i	Criterion	i	**Criterion**		(u, v)	Criterion	(u, v)	**Criterion**	
1	-1	33.79	-1	**33.79**	**31.39**	-	-	2	**34.79**	**34.56**	1,0	37.66	1,0	**37.66**	**41.42**
2	-2	46.16	-2	**21.70**	**21.87**	-	-	3	**14.58**	**12.69**	-1,2	43.84	-1,4	**9.50**	**10.18**
3	-3	48.98	-6	**20.65**	**23.87**	-	-	6	**9.88**	**8.51**	0,2	45.78	-1,5	**1.77**	**2.34**
4	5	49.20	5	**19.88**	**20.03**	-	-	7	**6.24**	**5.01**	1,2	48.76	-1,9	**0.39**	**1.34**
5	3	52.07	-5	**19.22**	**20.87**	-	-	5	**5.47**	**4.17**	-1,4	53.56	-2,3	**0**	**0.33**
6	-4	53.12	10	**18.83**	**19.70**	-	-	4	**5.19**	**4.01**	1,4	56.54	0,2	**0**	**0.17**
7	7	55.05	12	**18.17**	**19.03**	-	-	1	**5.47**	**4.01**	-1,3	57.59	-1,2	**0**	**0.17**
8	-6	57.81	15	**17.84**	**19.53**	-	-	-	-	-	2,2	58.97	1,4	**0**	**0.17**
9	4	58.92	-7	**17.67**	**19.53**	-	-	-	-	-	1,3	59.41	2,0	**0**	**0**
10	-5	59.80	13	**17.50**	**19.20**	-	-	-	-	-	2,0	59.75	1,2	**0**	**0**

As it can be expected, the Fourier descriptors are unable to achieve an error free classification. The objects with the nested holes $\{4, 9, 13\}$ are only considered with respect to their outside boundary, which is identical. Also the two symmetrical objects $\{5, 6\}$ contribute to the high error rate of the Fourier descriptors. The classification performance of the invariant moments is quite satisfactory. The highest estimated respectively detected accuracy is reached with six features. Once again the UNL Fourier Features are able to discriminate among all test objects without failure. Using the first 9 or 10 selected features yields an estimated and detected error of 0%. Also for the entire set of objects, the results are very encouraging.

In order to give an idea about the cluster qualities of the different 2-D shape descriptors, the two best features of each descriptor in Table 5.3 were used to visualize the samples. These are $\{F(-1), F(-2)\}$ for the Fourier descriptors, the invariant moments $\{\phi_2, \phi_3\}$ and the UNL Fourier Features $\{UFF(1,0), UFF(-1,4)\}$. The five objects $\{4, 5, 11, 12, 13\}$ of Figure 5.16 were randomly chosen from the total set of 13 objects. This reduction aims to avoid the overloading of the graph. Besides, the 50 samples of the *test set* were used, and not the 150 of the *training set*. Figure 5.19 shows the location of the test samples for the five objects in the 2-D feature space of the best two features.

Fourier Descriptors: This kind of 2-D shape descriptor shows a good ability to separate among the classes, except for the two objects $\{4, 13\}$, which are completely overlapping. The overlapping effect is absolutely consistent with reality, since the two objects have the same outside border. They are therefor identical from the point of view of the Fourier descriptors.

Invariant Moments: Class 5 forms an elongated cluster in the $\{\phi_2, \phi_3\}$-space, but distinguishes still linearly from the other classes. Classes 11 and 12 form a joint cluster near the origin of the coordinate system. They are too similar for this shape features to separate them from the other classes. The same complete overlapping happens for the classes $\{4, 13\}$. Also for the invariant moments, the differences of the two classes are too insignificant for classification.

UNL Fourier Features: The best clustering qualities are doubtlessly in the subspace of the two firstly selected features $\{UFF(1,0), UFF(-1,4)\}$. Classes $\{5, 11, 12\}$ form a perfectly separable cluster. The classes $\{4, 13\}$ however pose a problem to the classifier. The partial overlapping of the classes contributes surely to the non-zero classification error when only the best two features are used, refer to Table 5.3.

5.2.6.4 Experiment: Recognition of Single-writer Handwritten Digits

It will be shown that the UNL Fourier Features do not only have the ability to classify shapes which are typical for 2-D vision, but also shapes of a quite distinct nature, namely handwritten characters. These forms typically consist of a set of *strokes* that the writer makes to assemble the final character, see Figure 5.9 on page 13.

In [Rauber, 1992a] a classifier was presented that is exclusively based on UNL Fourier Features. The patterns are from the domain of the ten arabic digits, “0” to “9”. The acquisition device is a bitmap tablet. The points of the patterns are stored in a file and transferred to a module that extracts the UNL Fourier Features. The recognition is off-line. That signifies that only the static information of the pattern points is used for recognition. The temporal relationships of the character strokes are ignored and lost during the acquisition process.

Later on, a complete system will be described that aims at recognizing handwritten digits of a single writer. Various 2-D shape descriptors are merged into a single feature model. The philosophy of the classifier is to provide as much information as possible, and then select the most useful by means of feature selection techniques.

Figure 5.20 shows the schematic model of the pattern acquisition module for the handwritten digit recognition. The writer leads the pen over the surface of the digitizer tablet, a PREH “Bitmap 12” with a resolution of 2200×2200 pixels. The original resolution is linearly mapped to the resolution of the Cartesian coordinate system, in this case $N_C = 128$. The tablet is controlled by a PC and the incoming pixel coordinates are registered. Since the PC was not able to trace all pixels that are passed by the pen in real time, a linear interpolation between two incoming coordinates is performed. This avoids that gaps appear along the curves that compose the digit pattern. The coordinates of the patterns are then stored in a file and passed to the classifier module, where feature extraction, feature selection and training and test is performed.

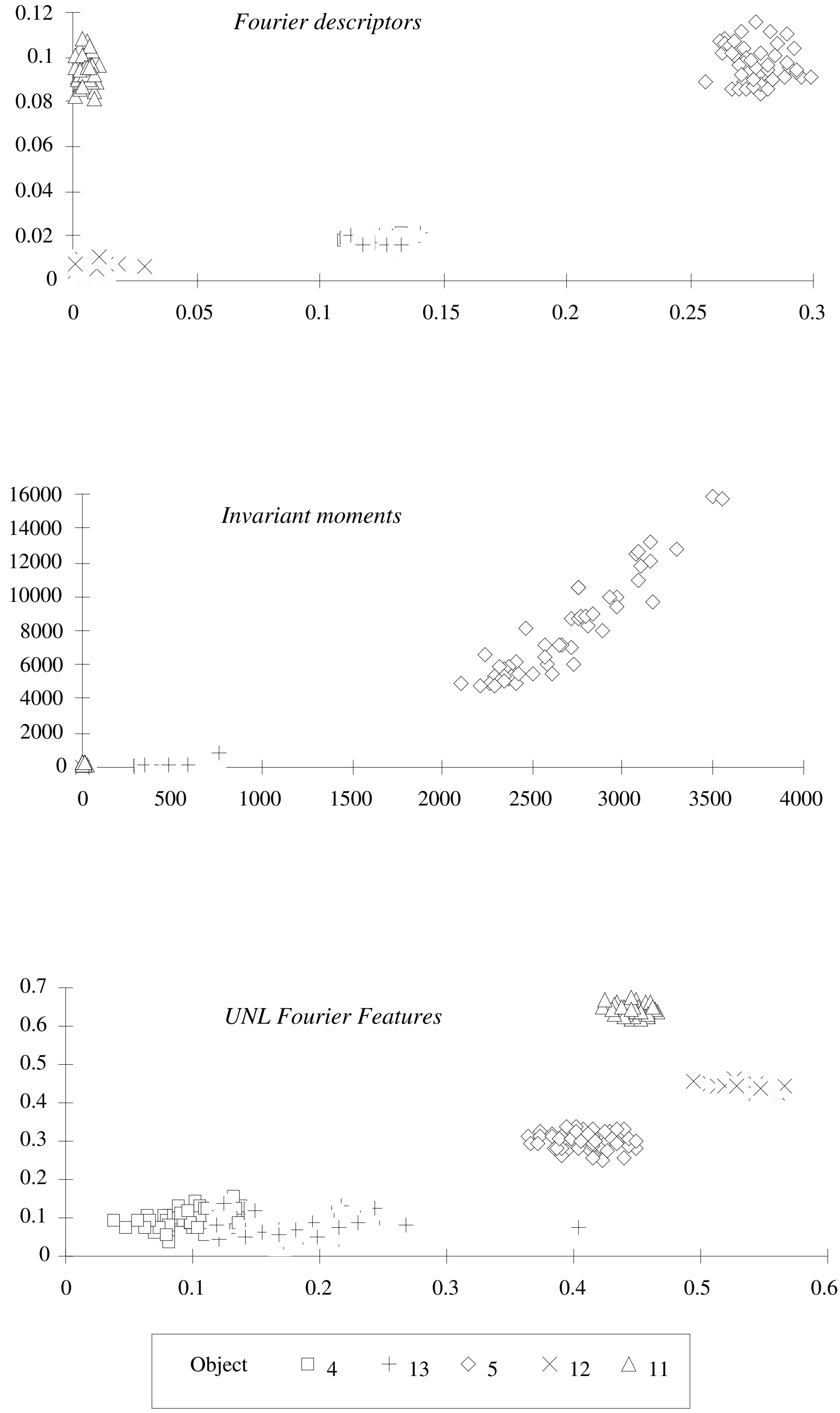


Figure 5.19 Class cluster in the 2-D feature subspace of the best two features.

The digits are all provided by a single writer, with a certain kind of writing discipline. To allow the reader to get an idea about the topology and variety of the patterns, the complete training and test set is shown in Annex A2. The test set is framed with a box in the figure. The number of training samples is about 100 for each class. The number of test samples is about 20 for each class.

It should be mentioned that the classifier is able to distinguish the two characters “6” and “9” because for the particular pattern set of Annex A2, the author uses two stroke patterns that are *not* rotation symmetric. The UNL Fourier Features *are* rotation invariant. Not only for this reason, a robust character recognition cannot only rely on this kind of features. Other approaches must be found which keep the general purpose recognition system not too much in a dependency of a specific writer’s style.

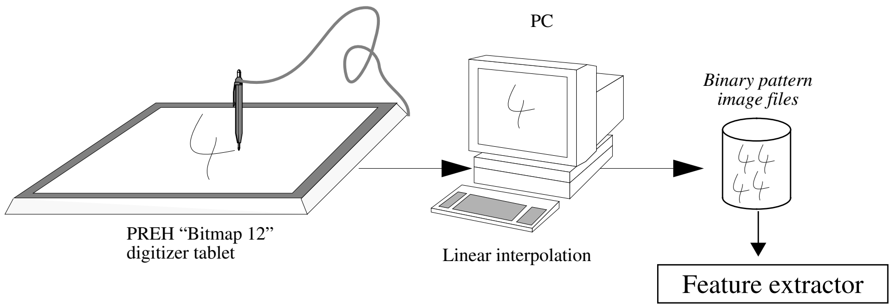


Figure 5.20 Pattern acquisition module for handwritten digit recognizer

The results of the classifier correspond to the methods of the anterior experiment. The whole feature set is preselected to a number of 100. Then a selection with the MINERR algorithm is performed. The best overall classification performance was achieved by using the first 10 selected features $UFF(u, v)$ with the indices (u, v) in the order $(1, 2)$, $(0, 1)$, $(1, 1)$, $(0, 2)$, $(1, 3)$, $(-2, 3)$, $(1, 0)$, $(-2, 1)$, $(-1, 2)$, $(2, 4)$. The estimated accuracy is 100% and the real accuracy for the test samples is 99.49%. The recognition rate is shown in Table 5.4.

It can be observed that the error estimation gives results that are too optimistic. The real recognition error when classifying the test set is generally higher than the estimated error. Nevertheless this classification experiment proves that the UNL Fourier Features are 2-D shape descriptors that are robust even in the presence of large amounts of non-physical noise, i.e. variations of the pattern caused by a human’s incapacity to reproduce exactly the same prototype pattern over and over again.

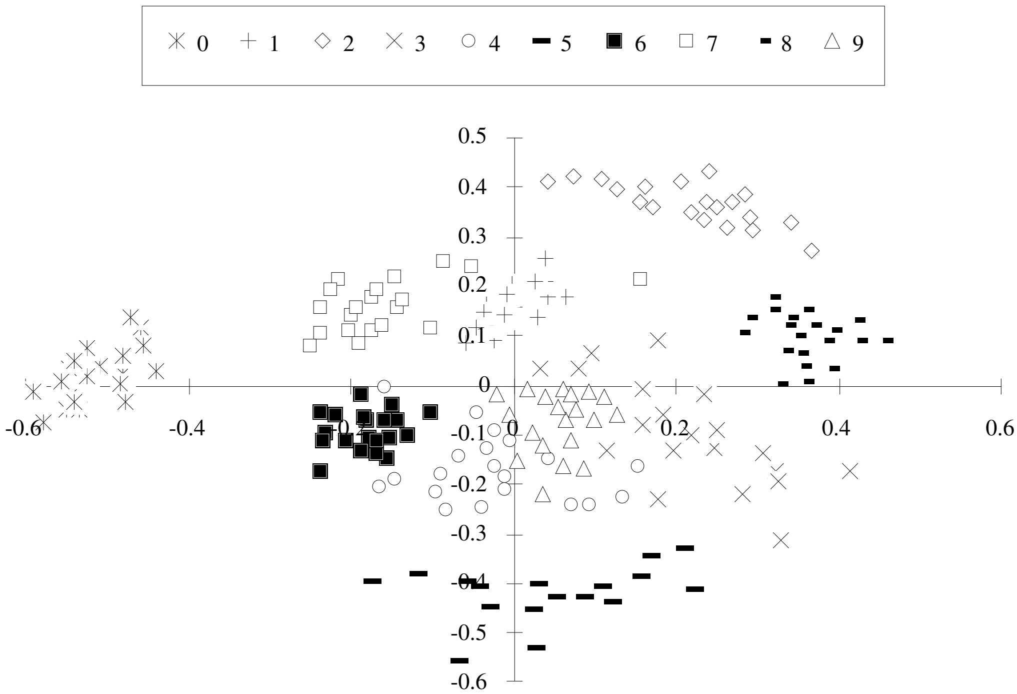


Figure 5.21 Sammon mapping of the best 10 UNL Fourier Features of the 10 digit classes.

In Figure 5.21 the discriminative properties of the UNL Fourier Features are illustrated in a graph that represents a Sammon's mapping [Sammon, 1969]. The number of iterations was set to 1000. The mapping permits a visualization of high dimensional data in just two dimensions. The graph of Figure 5.21 shows the clusters of the ten digits in the 10-dimensional space of the ten best features. It is clearly visible that the pattern classes form more or less compact clusters.

Table 5.4 Performance of a single-writer handwritten digit recognizer based exclusively on UNL Fourier Features

	RECOGNITION RATE [%]									
Nr. of features	1	2	3	4	5	6	7	8	9	10
Estimated	42.52	74.42	89.79	96.21	98.42	99.16	99.37	99.68	99.79	100
Detected	42.05	66.15	85.64	90.26	94.36	97.44	98.97	98.46	98.97	99.49

5.2.7 Concluding Remarks

In section 5.2 a novel method was presented for the representation of two-dimensional shapes. First, the domain of the method was defined, patterns that are composed of thin curves. All necessary definitions for parametric curves were given. Then a complete theory with a sound mathematical framework was presented. After the philosophy of the pattern synthesis, the normalizing procedures were presented. The translation, scale change and rotation to translation normalization is done by the analytical UNL transform. The final feature set is obtained by a 2-D Fourier transform. The essential qualities of a 2-D form descriptors were mathematically proved, namely insensitivity in relation to translation, scale changes and rotation of the original pattern.

After the theoretical framework, methods were presented to apply the UNL Fourier Features to real world pattern recognition problems. Since the analytic description of the pattern is mostly missing, approximative estimations must be made to perform the UNL transform of a discrete bitmap pattern. The problem of applying the discrete UNL transform was solved.

The method was compared to other 2-D shape descriptors, and its distinct nature was outlined.

Then in an experimental part, the ability to discriminate shapes was illustrated. Although from the particular benchmarks that were used as a test case for the UNL Fourier Features no generalizations can be made, the method proved its robustness with the recognition of real world objects and shapes. Not only patterns from typical vision environments are recognized, but also handwritten characters.

Annex A1 Proof of Lemma 5.3

To prove:

The UNL transform $U(\Omega(t))$ of the object $\Omega(t)$ is invariant to translations of $\Omega(t)$ by the offset Δz.

Proof:

A translation of an object $\Omega(t)$ to a new object $\Omega^*(t)$ is equivalent to adding an offset Δz to the object, respectively all curves that compose the object:

$$\Omega^*(t) = \Omega(t) + \Delta z \equiv \{z_i(t) + \Delta z | \, i = 1 \ldots n\}$$

We have to prove that $U(\Omega^*(t)) = U(\Omega(t))$.

The new centroid of the translated pattern O^* becomes $O^* = O + \Delta z$ which is proved in the following:

$$O^* = \frac{\sum_{i=1}^{n} \int_0^1 z_i^*(t) \left\| \dot{z}_i^*(t) \right\| dt}{\sum_{i=1}^{n} \int_0^1 \left\| \dot{z}_i^*(t) \right\| dt} \quad \text{with } z_i^*(t) = z_i(t) + \Delta z.$$

The derivative $d(z_i^*(t))/dt$ is equal to $d(z_i(t))/dt$ since the offset Δz is a constant additive term which disappears during derivation. With the condition that Δz does not depend on the integration variable t or on the index i of the curves we can write

$$O^* = \frac{\sum_{i=1}^{n} \int_0^1 z_i^*(t) \left\| \dot{z}_i^*(t) \right\| dt}{\sum_{i=1}^{n} \int_0^1 \left\| \dot{z}_i^*(t) \right\| dt} = \frac{\sum_{i=1}^{n} \int_0^1 (z_i(t) + \Delta z) \left\| \dot{z}_i(t) \right\| dt}{\sum_{i=1}^{n} \int_0^1 \left\| \dot{z}_i(t) \right\| dt}$$

$$= \frac{\sum_{i=1}^{n} \int_0^1 z_i(t) \left\| \dot{z}_i(t) \right\| dt}{\sum_{i=1}^{n} \int_0^1 \left\| \dot{z}_i(t) \right\| dt} + \frac{\Delta z \sum_{i=1}^{n} \int_0^1 \left\| \dot{z}_i(t) \right\| dt}{\sum_{i=1}^{n} \int_0^1 \left\| \dot{z}_i(t) \right\| dt} = O + \Delta z \qquad \text{q.e.d.}$$

The new maximum distance M^* remains invariant in relation to the old M:

$$M^* = max\{\left\| z_i^*(t) - O^* \right\|\} \qquad \forall i = 1 \ldots n \qquad t \in (0, 1)$$

$$= max\{\| z_i(t) + \Delta z - (O + \Delta z) \|\} = max\{\| z_i(t) - O \|\} = M \qquad \text{q.e.d.}$$

Finally, with $O = (O_x, O_y)$ and $\Delta z = (\Delta x, \Delta y)$ we show that $U(\Omega^*(t)) = U(\Omega(t))$:

$$U(z_i^*(t)) = \left(\frac{\left\| z_i^*(t) - O^* \right\|}{M^*}, \text{atan}\left(\frac{y_i^*(t) - O_y^*}{x_i^*(t) - O_x^*} \right) \right)$$

$$= \left(\frac{\| z_i(t) + \Delta z - (O + \Delta z) \|}{M}, \text{atan}\left(\frac{y_i(t) + \Delta y - (O_y + \Delta y)}{x_i(t) + \Delta x - (O_x + \Delta x)} \right) \right) = U(z_i(t)) \qquad \text{q.e.d.}$$

To prove:

The UNL transform $U(\Omega(t))$ of the object $\Omega(t)$ is invariant to scale changes by a scalar a.

Proof:

A scale change of the object $\Omega(t)$ to the new object $\Omega^*(t)$ is equivalent to the multiplication of each pattern curve by a scalar $a \in \boldsymbol{R}^+$:

$$\Omega^*(t) = a\Omega(t) \equiv \{az_i(t) \mid i = 1 \ldots n\}$$

Once again it is shown that $U(\Omega^*(t)) = U(\Omega(t))$.

The new centroid O^* becomes $O^* = aO$:

$$O^* = \frac{\sum_{i=1}^{n} \int_0^1 z_i^*(t) \| \dot{z}_i^*(t) \| dt}{\sum_{i=1}^{n} \int_0^1 \| \dot{z}_i^*(t) \| dt}$$

Since a is a scalar that is independent of t or i, we have $dz_i^*(t)/dt = a(dz_i(t)/dt)$. Furthermore $\| a\dot{z}_i(t) \| = a \| \dot{z}_i(t) \|$ for all $a \in \boldsymbol{R}^+$.

$$O^* = \frac{\sum_{i=1}^{n} \int_0^1 az_i(t) a \| \dot{z}_i(t) \| dt}{\sum_{i=1}^{n} \int_0^1 a \| \dot{z}_i(t) \| dt} = aO \qquad \text{q.e.d.}$$

The new maximum distance M^* becomes $M^* = aM$:

$$M^* = max\{\| z_i^*(t) - O^* \|\} \qquad \forall i = 1 \ldots n \qquad t \in (0, 1)$$

$$= max\{\| az_i(t) - aO \|\} = max\{a \| z_i(t) - O \|\} = a(max\{\| z_i(t) - O \|\}) = aM \qquad \text{q.e.d.}$$

With $az(t) = a(x(t), y(t)) = (ax(t), ay(t))$ and $aO = a(O_x, O_y) = (aO_x, aO_y)$ we obtain

$$U(z_i^*(t)) = \left(\frac{\| z_i^*(t) - O^* \|}{M^*}, \operatorname{atan}\left(\frac{y_i^*(t) - O_y^*}{x_i^*(t) - O_x^*} \right) \right)$$

$$= \left(\frac{\| az_i(t) - aO \|}{aM}, \operatorname{atan}\left(\frac{ay_i(t) - aO_y}{ax_i(t) - aO_x} \right) \right) = U(z_i(t)) \qquad \text{q.e.d.}$$

To prove:

A rotation about any point $z_r = (x_r, y_r)$ by the angle $\Delta\theta$ has no effect on the first variable $R(t)$ of the UNL transform and causes a 2π-periodic translation of the second variable $\theta(t)$ by $\Delta\theta$ (cyclic shift).

Proof:

It remains to prove that rotations of the original pattern result in cyclic shifts of the UNL transformed pattern. The cyclic shifts occur in the $\theta(t)$ parameter of the UNL transform. The $R(t)$ remains invariant.

Generally, a rotation of an object, respectively all points that make up the object around an arbitrary point p can be decomposed into v.) A translation Δz of the object, such that the centroid O matches the rotation point z_r

vi.) A rotation of the object around its centroid O by an angle $\Delta\theta$

vii.) An inverse translation $-\Delta z$ of the object

It will be proved that $U(z_i^*(t)) = (R_i^*(t), \theta_i^*(t)) = (R_i(t), \theta_i(t) + \Delta\theta)$

The rotation of a pattern curve $z_i(t)$ or any single coordinate by an angle $\Delta\theta$ is equivalent to multiplying it by the complex term $e^{j\Delta\theta}$: $z_i^*(t) = z_i(t)e^{j\Delta\theta}$.

The new values for O and M are calculated as follows:

Since $dz_i^*(t)/dt = d(z_i(t)e^{j\Delta\theta})/dt = e^{j\Delta\theta}(dz_i(t)/dt)$ and $\| e^{j\theta} \| = 1$ for all angles θ, we have

$$O^* = \frac{\sum_{i=1}^{n} \int_0^1 z_i^*(t) \| \dot{z}_i^*(t) \| dt}{\sum_{i=1}^{n} \int_0^1 \| \dot{z}_i^*(t) \| dt} = \frac{\sum_{i=1}^{n} \int_0^1 z_i(t)e^{j\Delta\theta} \| \dot{z}_i(t)e^{j\Delta\theta} \| dt}{\sum_{i=1}^{n} \int_0^1 \| \dot{z}_i(t)e^{j\Delta\theta} \| dt} = Oe^{j\Delta\theta}$$

Ignore that the rotation is done around O, i. e. $O = O^*$

$$M^* = max\{ \| z_i^*(t) - O^* \| \} \qquad \forall i = 1\ldots n \qquad t \in (0, 1)$$

$$= max\{ \| z_i(t)e^{j\Delta\theta} - Oe^{j\Delta\theta} \| \} = max\{ \| (z_i(t) - O)e^{j\Delta\theta} \| \} = max\{ \| z_i(t) - O \| \} = M$$

A rotation in 2-D of a point (x, y) about an angle φ to a new point (x^*, y^*) is equivalent to the transformations:

$$x^* = x\cos\varphi - y\sin\varphi \qquad y^* = x\sin\varphi + y\cos\varphi$$

The following trigonometric relation for the tangents of two angles α and β holds:

$\tan(\alpha + \beta) = \dfrac{\tan\alpha + \tan\beta}{1 - \tan\alpha\tan\beta}$. Hence we can prove that $U(z_i^*(t)) = (R_i(t), \theta_i(t) + \Delta\theta)$.

$$U(z_i^*(t)) = \left(\frac{\| z_i^*(t) - O^* \|}{M^*}, \operatorname{atan}\left(\frac{y_i^*(t) - O_y^*}{x_i^*(t) - O_x^*} \right) \right)$$

$$= \left(\frac{\| z_i(t)e^{j\Delta\theta} - Oe^{j\Delta\theta} \|}{M}, \operatorname{atan}\left(\frac{x_i(t)\sin\Delta\theta + y_i(t)\cos\Delta\theta - (O_x\sin\Delta\theta + O_y\cos\Delta\theta)}{x_i(t)\cos\Delta\theta - y_i(t)\sin\Delta\theta - (O_x\cos\Delta\theta - O_y\sin\Delta\theta)} \right) \right)$$

$$= \left(\frac{\| (z_i(t) - O)e^{j\Delta\theta} \|}{M}, \operatorname{atan}\left(\frac{(y_i(t) - O_y)\cos\Delta\theta + (x_i(t) - O_x)\sin\Delta\theta}{(x_i(t) - O_x)\cos\Delta\theta - (y_i(t) - O_y)\sin\Delta\theta} \right) \right)$$

$$= \left(\frac{\| z_i(t) - O \| \| e^{j\Delta\theta} \|}{M}, \operatorname{atan}\left(\frac{(y_i(t) - O_y)\cos\Delta\theta + (x_i(t) - O_x)\sin\Delta\theta}{(x_i(t) - O_x)\cos\Delta\theta - (y_i(t) - O_y)\sin\Delta\theta} \frac{(x_i(t) - O_x)\cos\Delta\theta}{(x_i(t) - O_x)\cos\Delta\theta} \right) \right)$$

$$= \left(\frac{\| z_i(t) - O \|}{M}, \operatorname{atan}\left(\frac{\dfrac{y_i(t) - O_y}{x_i(t) - O_x} + \dfrac{\sin\Delta\theta}{\cos\Delta\theta}}{1 - \dfrac{y_i(t) - O_y}{x_i(t) - O_x}\dfrac{\sin\Delta\theta}{\cos\Delta\theta}} \right) \right) = \left(\frac{\| z_i(t) - O \|}{M}, \operatorname{atan}(\tan((z_i(t) - O) + \Delta\theta)) \right)$$

$$= (R_i(t), (z_i(t) - O) + \Delta\theta) = (R_i(t), \operatorname{atan}(\tan(z_i(t) - O)) + \Delta\theta)$$

$$= \left(R_i(t), \operatorname{atan}\left(\frac{y_i(t) - O_y}{x_i(t) - O_x} \right) + \Delta\theta \right) = (R_i(t), \theta_i(t) + \Delta\theta) \qquad \text{q.e.d.}$$

A rotation provokes a cyclic shift with a period of 2π. Since $\theta_i(t)$ is a parametric angle function, the 2π-periodicity of $\theta_i(t)$ is true by nature.

Annex A2 Pattern Set "Digits"

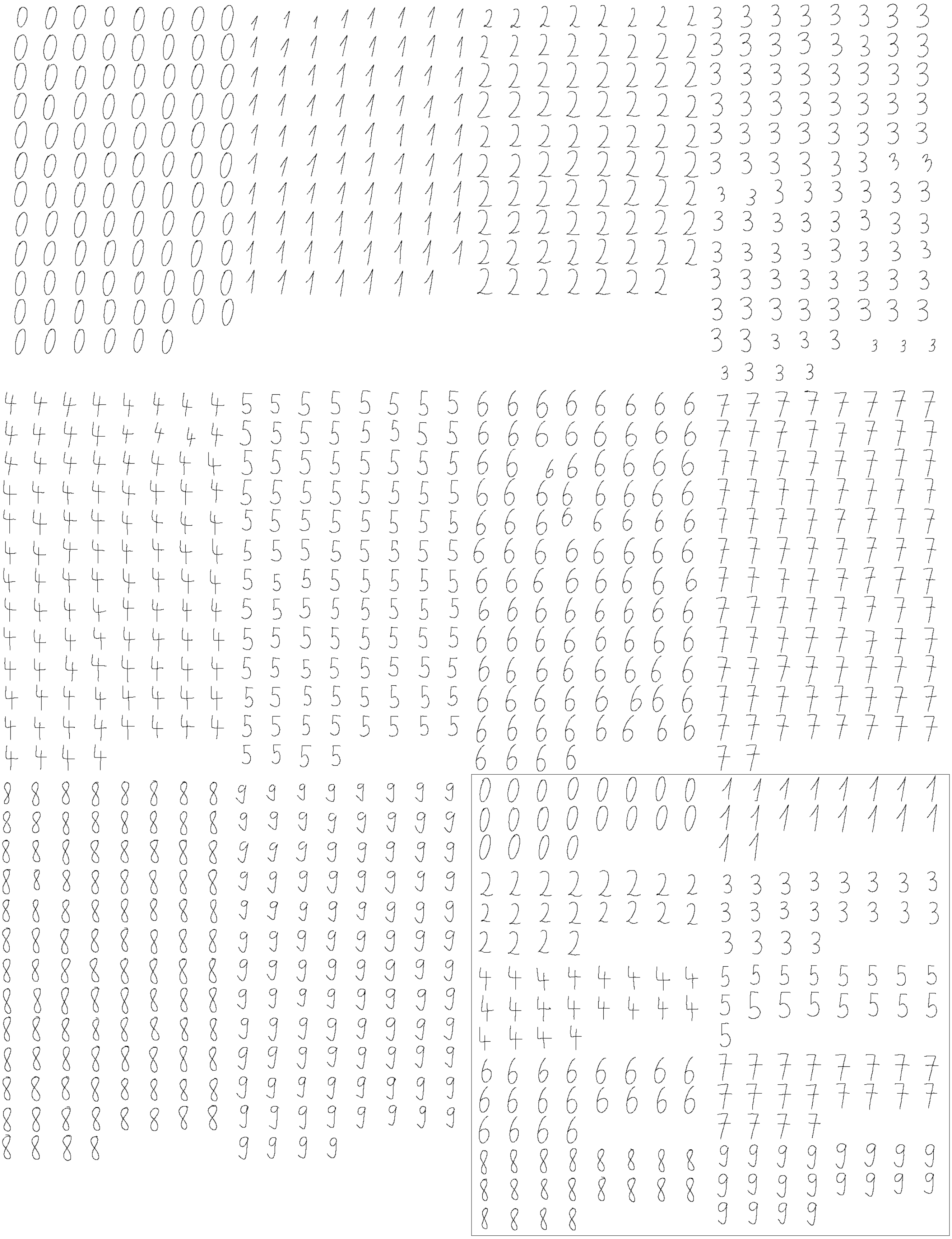


Figure A.1 The pattern set of the handwritten digits recognition based on UNL Fourier Features (test patterns in box).

BIBLIOGRAPHY

[Burr, 1986]

Burr, D. J., "A neural network digit recognizer," *Proc. IEEE Conf. Systems, Man, and Cybernetics*, Atlanta, GA, pp. 1621-1625, Oct. 1986.

[Burr, 1988]

Burr, D. J., "Experiments on neural network recognition of spoken and written text," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1162-1168, Jul. 1988.

[Devijver, 1982]

Devijver, P. A., and Kittler, J., *Pattern Recognition: A Statistical Approach*, Prentice/Hall Int., London, 1982.

[Freeman, 1961]

Freeman, H., "On the encoding of arbitrary geometric configurations," *IEEE Trans. on Electronic Computers*, vol. EC-10, pp. 260-268, 1961.

[Fu, 1987]

Fu, S., Gonzalez, R. C., and Lee, C. S. G., *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, New York, 1987.

[Gonzalez, 1987]

Gonzalez, R. C., and Wintz, P., *Digital Image Processing (2nd ed.)*, Addison-Wesley, 1987.

[He, 1991]

He, Y., and Kundu, A., "2-D Shape Classification Using Hidden Markov Model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1172-1184, Nov. 1991.

[Hu, 1962]

Hu, M.-K., "Visual pattern recognition by moment invariants," *IRE Trans. Information Theory*, vol. IT-8, pp. 179-187, Feb. 1962.

[Khotanzad, 1990]

Khotanzad, A., and Hong, Y. H., "Invariant image recognition by Zernike moments," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489-497, May 1990.

[Massone, 1985]

Massone, L., Sandini, G., and Tagliasco, V., "'Form-invariant' topological mapping strategy for 2D shape recognition," *Computer Vision, Graphics and Image Processing*, vol. 30, pp. 169-188, 1985.

[Nasrabadi, 1991]

Nasrabadi, N. M., and Li, W., "Object recognition by a Hopfield neural network," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 21, no. 6, pp. 1523-1535, Nov./Dec. 1991.

[Pavlidis, 1977a]

Pavlidis, T., "Polygonal approximations by Newton's method," *IEEE Trans. Computers*, vol. C-26, no. 8, pp. 800-807, Aug. 1977.

[Pavlidis, 1977b]

Pavlidis, T., *Structural Pattern Recognition*, Springer-Verlag, New York, 1977.

[Pavlidis, 1978]

Pavlidis, T., "SURVEY-A review of algorithms for shape analysis," *Computer Graphics and Image Processing*, vol. 7, pp. 243-258, 1978.

[Pavlidis, 1982]

Pavlidis, T., *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD, 1982.

[Persoon, 1977]

Persoon, E., and Fu, K.-S., "Shape discrimination using Fourier Descriptors," *IEEE*

Trans. Systems, Man, and Cybernetics, vol. SMC-7, no. 3, pp. 170-179, 1977.

[Rauber, 1992a]

Rauber, T. W., and Steiger-Garção, A. S., “Shape description by UNL Fourier features—An application to handwritten character recognition,” *Proc. 11th IAPR Int. Conf. on Pattern Recognition*, The Hague, The Netherlands, vol. II, pp. 466-469, Aug. 1992.

[Rauber, 1992b]

Rauber, T. W., and Steiger-Garção, A. S., “2-D form descriptors based on a normalized parametric polar transform (UNL Transform),” *Proc. MVA’92 – IAPR Workshop on Machine Vision Applications*, NEC Super Tower, Tokyo, Japan, Dec. 7-9, 1992.

[Reber, 1987]

Reber, W. L., and Lyman, J., “An artificial neural system design for rotation and scale invariant pattern recognition,” in *Proceedings of IEEE First International Conference on Neural Networks*, San Diego, CA, vol. IV, pp. 277-283, 1987.

[Reeves, 1988]

Reeves, A. P., Prokop, R. J., Andrews, S. E., and Kuhl, F. P., “Three-dimensional shape analysis using moments and Fourier descriptors,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 937-943, Nov. 1988.

[Reiss, 1991]

Reiss, T. H., “The revised fundamental theorem of moment invariants,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 830-834, Aug. 1991.

[Sammon, 1969]

Sammon, J. W., Jr., “A nonlinear mapping for data structure analysis”, *IEEE Trans. on Computers*, vol. C-18, no. 5, pp. 401-409, May 1969.

[Schwartz, 1977]

Schwartz, E. L., “Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception,” *Biological Cybernetics*, vol. 25, pp. 181-194, 1977.

[Schwartz, 1980a]

Schwartz, E. L., “A quantitative model of the functional architecture of human striate cortex with applications to visual illusion and cortical texture analysis,” *Biological Cybernetics*, vol. 37, pp. 63-76, 1980.

[Schwartz, 1980b]

Schwartz, E. L., “Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding,” *Vision Research*, vol. 20, pp. 645-669, 1980.

[Teh, 1988]

Teh, C.-H., and Chin, R. T., “On image analysis by the methods of moments,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 496-513, July 1988.

[Wechsler, 1988]

Wechsler, H., and Zimmerman, G. L., “2-D Invariant Object Recognition Using Distributed Associative Memory,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-10, no. 6, pp. 811-821, Nov. 1988.

[Wolfram, 1991]

Wolfram, S., *Mathematica: A System for Doing Mathematics by Computer (2nd ed.)*, Addison-Wesley Publishing Company, 1991.

[Zahn, 1972]

Zahn, C. T., and Roskies, R. Z., “Fourier descriptors for plane closed curves,” *IEEE Trans. on Computers*, vol. C-21, no. 3, pp. 269-281, 1972.

[Zhang, 1984]

Zhang, T. Y., and Suen, C. Y., “A fast parallel algorithm for thinning digital patterns”, Communications of the ACM, vol. 27, no. 3, pp. 236-239, 1984.